

UNIVERSITATEA POLITEHNICA BUCUREȘTI  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE



## PROIECT DE DIPLOMĂ

### **Aplicație web pentru managementul învățării** versiunea 2024

Panait Ștefan-Andrei

**Coordonator științific:**  
Prof. dr. ing. Andrei Ionescu

**BUCUREȘTI**

2024

UNIVERSITY POLITEHNICA OF BUCHAREST  
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS  
COMPUTER SCIENCE DEPARTMENT



## DIPLOMA PROJECT

Aplicație web pentru managementul învățării  
2024 version

Ioana Popescu

**Thesis advisor:**  
Prof. dr. ing. Andrei Ionescu

**BUCHAREST**

2024

# 1 CONTENTS

Sinopsis .....	3
Mulțumiri .....	4
1 Introducere .....	5
1.1 Context .....	5
1.2 Problema .....	5
1.3 Obiective .....	6
1.4 Structura lucrării.....	7
2 Analiza și specificarea cerințelor.....	8
2.1 Cerinte nonfunctionale .....	9
2.2 Cerinte functionale.....	9
3 Studiu de piață / Abordări existente.....	13
4 Soluția propusă .....	21
4.1 Spring Boot .....	21
4.1.1 Acces la date .....	<b>Error! Bookmark not defined.</b>
4.1.2 Securitate .....	<b>Error! Bookmark not defined.</b>
4.2 Angular .....	21
4.3 MySQL .....	22
4.4 Open-Source.....	22
4.5 Arhitectura bazei de date.....	23
4.6 Docker .....	24
4.7 Principiul dezvoltare continua si livrare continua.....	24
4.8 Descrierea fluxului funcționalităților .....	25
5 Detalii de implementare .....	28
5.1 Autentificare si securitate .....	28
5.1.1 Autentificare pe baza cookie-urilor .....	28
5.1.2 OAuth 2.0 .....	29
5.1.3 JWT token .....	29
5.2 Accesul la date.....	33
5.2.1 JDBC Template .....	33
5.2.2 MyBatis .....	33

5.2.3	Spring Data JPA .....	33
5.2.4	Implementarea JPA .....	34
5.3	Paginare si cautare dupa diferite informatii .....	36
5.3.1	Detalii de implementare .....	37
6	Studiu de caz / Evaluarea rezultatelor .....	39
7	Concluzii .....	43
8	Bibliografie .....	44
9	Anexe .....	46

## **SINOPSIS**

Aplicația web pentru managementul învățării este o platformă ușor de utilizat pentru a facilita educației și a studiul suplimentar, online. Această are o interfață prietenoasă fiind foarte fiabilă atât pentru profesori cât și pentru cursanți în vederea interacționării, gestionării cursurilor. Cursanții înscriși pot să primească recenzii personalizate și să acceseze diferite resurse materiale educative.

Aplicația oferă siguranța utilizatorilor prin autentificare securizată și controlul accesului pe baza de rol pentru modificarea permisiunilor la funcționalități. Printre alte opțiuni, utilizatorul își poate personaliza profilul, atât în rolul de student cât și de instructor. Profesorii pot edita modulele cursurilor și emblema, pot da teme suplimentare și materiale elevilor, precum și recenzii personalizate și nota după verificarea temelor efectuate de cursanți.

## **MULȚUMIRI**

(opțional) Aici puteți introduce o secțiunea specială de mulțumiri / acknowledgments.

## **1 INTRODUCERE**

În domeniul educației, nevoia de inovație este constantă. Atât contextul mondial cât și tehnologia dezvoltată din ultimii ani au atins inevitabil și anii de temelie din viața fiecărui individ, mai exact, anii de școală astfel, modernizarea învățământului reprezintă o prioritate. Cu o tranziție neprevăzută și foarte rapidă în domeniul învățământului, profesorii și elevii au fost și sunt nevoiți să utilizeze diferite platforme online pentru continuarea predării și a studiului. Este evident faptul că nevoia de performanță, flexibilitate, cooperare și accesibilitate nu a fost niciodată la fel de mare.

Aplicația de management al învățării pe care o prezint se adresează acestor nevoi, oferind un mediu online complex pentru facilitarea și sprijinul educației și al dezvoltării. Această platformă este creată pentru îmbunătățirea experienței educaționale a studenților, a profesorilor, dar și a unităților de învățământ.

La bază, aplicația oferă o interfață centralizată, prietenoasă cu utilizatorii și clară, care favorizează comunicarea dintre profesori și cursanți. Cu capacități ce variază de la crearea cursului, gestionarea și editarea acestuia până la livrarea de conținut educațional suplimentar și evaluare detaliată și notarea temelor efectuate și trimise profesorului prin intermediul platformei, aplicația este de un real ajutor în interacțiunea profesor-cursant și facilitarea înțelegerii noțiunilor precum și suplimentarea orelor standard. Focalizându-se pe accesibilitate și performanță la nivel înalt, aplicația web asigură utilizatorilor navigarea cu ușurință pe platformă, accesarea resurselor materiale. Astfel, aplicația asigură elevilor siguranța că educația lor nu este afectată și pot continua călătoria lor de dezvoltare constant, indiferent de factorii care pot părea că împiedică continuitatea educației.

De altfel, consider că fiabilitatea și ușurința folosirii aplicației pot fi cele mai importante atuuri deoarece astfel poate fi la îndemână oricui, fără a fi nevoie de timp suplimentar special pentru înțelegerea și învățarea folosirii aplicației.

### **1.1 Context**

Situațiile neprevăzute din ultimii ani au dus către o schimbare drastică a organizării învățământului, de la metodele tradiționale de educație, la un nou tip de educație, cel online, care încă tinde să se dezvolte din ce în ce mai mult. Această schimbare a evidențiat importanța și nevoia unei platforme digitale bine dezvoltată ușor de folosit în vederea asigurării continuității educației.

### **1.2 Problema**

Învățarea suplimentară poate deveni greu de realizat în contextul tradițional în care profesorii și elevii sunt constrânși și nu își pot desfășura orele și materialele după cum doresc adaptându-se în funcție de fiecare elev în parte. În continuare voi enumera o parte din inconveniente care restricționează învățarea suplimentară în contextul tradițional:

- Limitările de timp
- Limitările de spațiu

- Numarul redus al resurselor materiale puse la dispozitie
- Lipsa interactivitatii si modernizarii
- Lipsa recenziilor detaliate si personalizate
- Lipsa flexibilitatii
- Lipsa temelor personalizate in functie de nevoile elevilor
- Constrangerea folosirii cartilor impuse de scoala

### 1.3 Obiective

Obiectivul principal pe care mi l-am propus in crearea unei noi platforme de invatare, in scopul invatarii suplimentare, il constituie facilitarea, sprijinirea, inovarea si imbunatatirea calitatii invatarii suplimentare, atat pentru studenti cat si pentru profesori.

Consider ca invatarea suplimentara trebuie sa fie la indemana oricui si usor de accesat si folosit, de aceea vdoresc sa evidentiez o parte dintre beneficiile pe care platforma le ofera profesorilor si studentilor.

#### a) Beneficii pentru profesor:

1. Flexibilitate in adaugarea resurselor materiale educative: Profesorii pot incarca oricand materialele educative care le sunt disponibile cursantilor in orice moment. De asemenea profesorii pot adauga materiale selectate de ei, nu doar din cartile de la scoala.
2. Crearea si gestionarea temelor: Profesorii pot adauga diferite probleme in lista de exercitii, de asemenea acestia pot select orice problema din lista pentru a crea o tema pentru elevi. Temele pot fi personalizate pentru nevoile comune ale mai multor studenti sau personalizate pentru un anumit elev. Profesorul poate impune sau nu un anumit termen limita pentru realizarea si incarcarea temelor de catre elevi.
3. Evaluarea temelor: Dupa efectuarea si incarcarea temelor de catre elevi, profesorii le pot corecta oricand. Corectarea temelor se poate realizeaza prin citirea lor si oferirea de recenzii personalizate fiecarui elev, cu exemple specifice din tema sa, incluzand si nota.

#### b) Beneficii pentru studenti:

1. Flexibilitatea in accesarea materialelor educationale oferite de profesori: Studentii pot vizualiza si descarca resursele materiale oferite de profesori si disponibile in orice moment.
2. Incarcarea temelor: Elevii pot incarca temele efectuate asigurandu-se ca lucrarile lor urmeaza sa fie evaluate de profesori si sa primeasca comentarii clare si precise cu greselile facute si sugestiile primite de catre cadrele didactice. De asemenea elevii dispun de flexibilitatea si in efectuarea si incarcarea pe platforma a temelor rezolvate.
3. Evaluare: Dupa evaluare temelor, elevii pot citi notitele si sugestiile profesorilor in timp ce isi pot revedea lucrarea, cat si rezolvarea corecta a acesteia. De asemenea fiecare elev primeste nota si recenzia in mod privat.

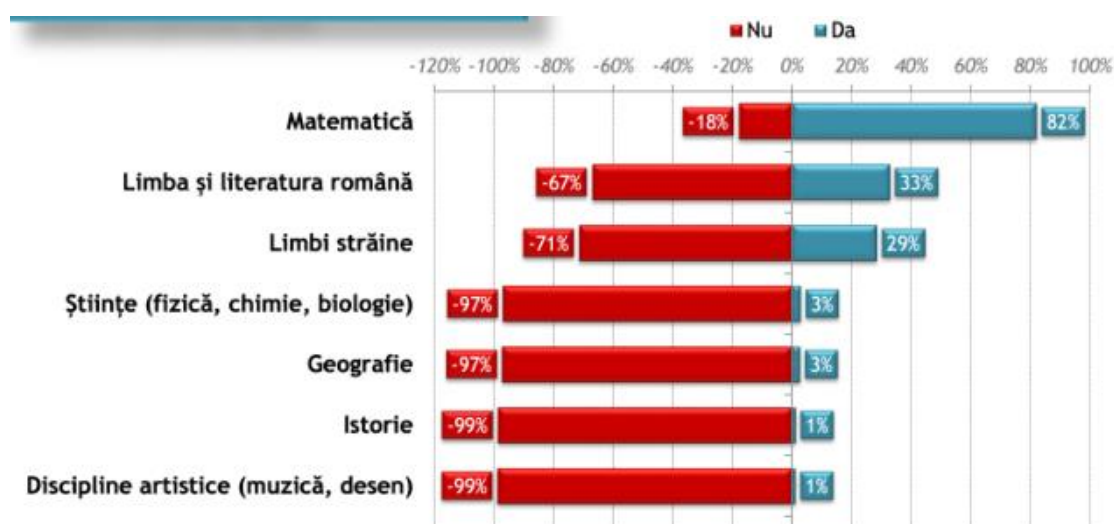


#### **1.4 Structura lucrării**

Un paragraf în care fiecare dintre secțiunile următoare este prezentată în 1-2 fraze, punând accentul pe elementele cele mai semnificative din fiecare secțiune.

## 2 ANALIZA ȘI SPECIFICAREA CERINȚELOR

Putem remarca faptul ca in Romania cultura meditatiilor este una foarte dezvoltata, fiecare parinte isi trimite copilul la meditatie macar in cei mai importanti ani din ciclul de invatamant al copiilor acestora. Un studiu la vinel national realizat de IRES si Societatea Academica din Romania spune ca „Conform studiului realizat de IRES și SAR în rândul părinților, o treime dintre elevi participă la sistemul de meditații, cu o frecvență considerabilă în clasele terminale, atunci când examenele naționale se apropie. În urma chestionării elevilor de gimnaziu și liceu, a rezultat faptul că un sfert dintre elevii din România accesează sistemul de meditații la cel puțin o materie.”<sup>1</sup>



Figură 1: Grafic ce reprezinta la ce materii au elevii nevoie de meditatie

Comparand aceste informatii cu alte tari:

- In Italia 1 din 5 elevi fac meditatie conform unei statistici realizate pe 6000 de elevi de repetizioni.it, platforma italiana pentru mediatii atat online cat si fizic.<sup>2</sup>
- In germania la fel ca in italia 1 din 5 elevi se adreseaza profesorilor pentru meditatie in invatamantul liceal conform lucrarii stiintifice „Effectiveness of private tutoring during secondary schooling in Germany: Do the duration of private tutoring and tutor qualification affect school achievement?”<sup>3</sup>

<sup>1</sup> <https://www.edupedu.ro/piata-meditatiilor-estimata-la-pest-120-de-milioane-de-lei-pe-luna-pe-baza-datelor-publice-si-a-cercetarilor-recente-analiza/>

<sup>2</sup> <https://www.rainews.it/articoli/2024/04/scuola-1-studente-su-5-va-a-ripetizioni-cosi-le-famiglie-spendono-circa-450-euro-7a461864-a445-4a71-849c-ff05fbf738b6.html#:~:text=Il%20trend%20C3%A8%20confortante%2C%20sia,si%20C3%A8%20passati%20al%2024%25>

<sup>3</sup> <https://www.sciencedirect.com/science/article/pii/S0959475219303093>

Din aceste statistici putem concluziona ca studentilor nu le sunt de ajuns orele de pregatire din cadrul scolilor. Astfel aceasta platforma ar actiona ca un pilon educational pentru student.

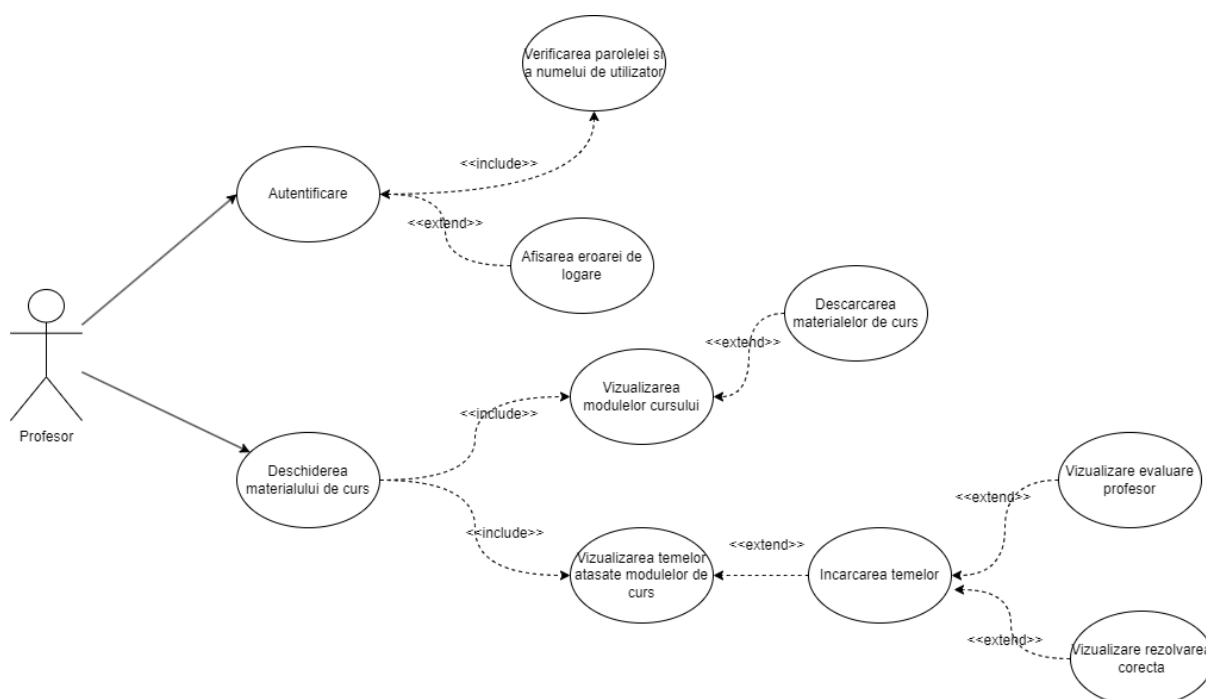
## 2.1 Cerinte nonfunctionale

Cerintele nonfunctionale se refera la specificatiile care descriu cum ar trebui sa se comporte un sistem, acestea se concentreaza pe calitatea si performanta aplicatiei si sunt esentiale pentru a asigura o experienta optima pentru utilizatori.

1. Securitate: datele utilizatorilor sunt protejate la orice moment de timp
2. Performanta: timpii de raspuns si viteza aplicatiei sa nu fie un impediment in folosirea acesteia
3. Viteza: Aplicatia trebuie sa raspunda cerintelor intr-un timp optim.
4. Fiabilitate: Aplicatia v-a functiona fara erori care sa afecteze disponibilitatea aplicatiei.

## 2.2 Cerinte functionale

Cerintele functionale se axeaza pe cum aplicatia functioneaza daca atinge sau nu anumite caracteristici dorite. In aplicatia propusa sunt 3 roluri: student, profesor si administrator. In continuare voi prezenta procesul de folosire a aplicatiei atasand diagrame ale cazurilor de utilizare.



Figură 2: Diagrama de utilizare a unui utilizator in rolul de student

In rolul de student procesul este urmatorul:

1. Creerea contului: Oricarei creare de cont nou ii este atribuit automat rolul de student.

2. Autentificarea: La autentificare studentul isi adauga credentialele urmand ca acestea sa fie verificate. Daca acestea sunt corecte utilizatorul este redirectat pe pagina unde sunt vizibile cursurile. Daca credentialele sunt incorecte acestuia ii va fi afisat un mesaj de eroare.
3. Inrolare la cursuri: Utilizatorul se poate inrola singur la cursurile la care crede ca are nevoie de studiu suplimentar.
4. Deschiderea materialului de curs: Dupa inrolarea la curs este accesibila deschiderea si privirea materialelor disponibile.
5. Vizualizarea modulelor cursului: Dupa deschiderea materialului de curs sunt vizibile modulele cursului. Un exemplu de modul: Titlul: „Polinoame”, Descriere: „Acest modul te va ajuta sa intelegi polinoamele” si un fisier atasat Suport curs: polinoame.pdf.
6. Descarcarea suportului de curs: Studentul poate descarca suportul de curs.
7. Vizualizarea temelor atasate modulelor de curs: Pentru fiecare modul cursantul poate avea atasat o tema.
8. Incarcarea temelor: Dupa ce studentul considera ca tema este rezolvata acesta poate incarca tema pentru evaluarea unui profesor.
9. Vizualizare evaluare profesor: Elevul poate observa comentariile profesorului asupra temei incarcate.
10. Vizualizare rezolvare corecta: Dupa evaluarea temei de catre profesor rezolvarea corecta a exercitiului este disponibila elevului.

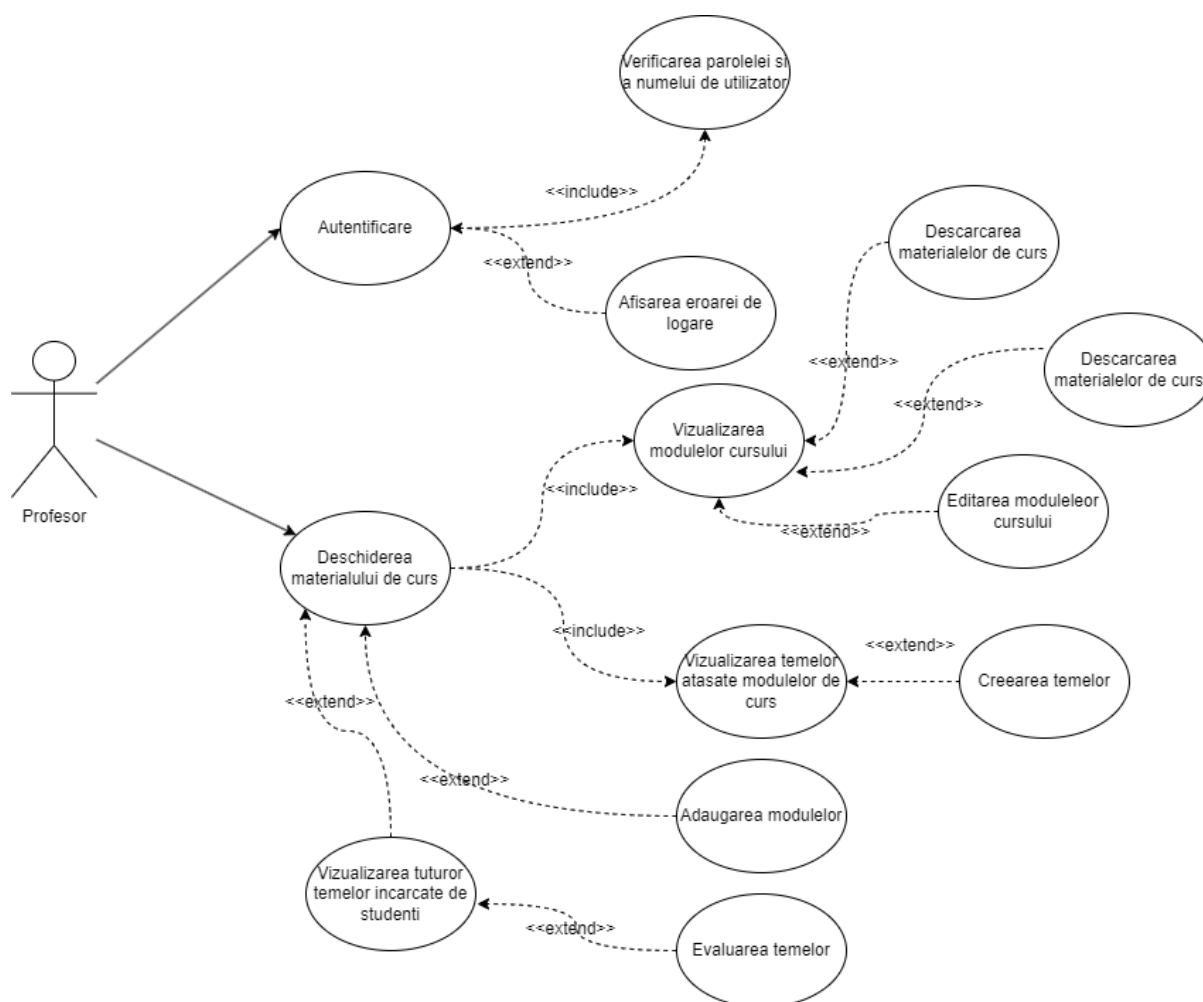


Figură 3: Diagrama de utilizare a unui utilizator in rolul de administrator

In rolul de administrator procesul este urmatorul:

1. Crearea de utilizator: administratorul poate crea utilizatori
2. Modificarea informatiilor utilizatorilor cat si a drepturilor de student sau profesor: administratorul poate schimba informatiile utilizatorului de ex numele, rolul sau adresa de email.
3. Stergerea de utilizatori
4. Crearea de cursuri: administratorul poate crea cursuri
5. Modificarea informatiilor cursurilor: Administratorul poate schimba titlul, descrierea cat si categoria din care face parte cursul
6. Stergerea de cursuri
7. Inrolarea de studenti sau profesori la cursuri: Elevul se poate inrola si singur la cursuri dar profesorul se poate inrola doar daca este inrolat de administrator. Ratiunea din spatele acestei functionalitati este ca doar un administrator poate stii daca un profesor este bun la o materie. Astfel nu se v-a intampla de exemplu ca un profesor de Informatica sa corecteze si sa puna materiale la Geografie.

8. Vizualizare lista de probleme: Administratorul poate vizualiza lista de probleme, acesta poate edita problemele sau de asemenea sterge probleme.
9. Adaugare de probleme: Administratorul de asemenea poate adauga probleme.



Figură 4: Diagrama de utilizare a unui utilizator in rolul de profesor

In rolul de profesor procesul este urmatorul:

1. Editarea siglei de curs: profesorul poate schimba sigla la cursurile la care acesta este inrolat.
2. Adaugarea de probleme in lista de probleme: profesorul poate adauga probleme
3. Adaugarea modulelor cursurilor: acesta poate adauga module noi pentru curs
4. Editarea modulelor de curs: Acesta poate edita titlul modulului, descrierea, cat si poate sterge sau incarca suportul de curs aferent modulului.
5. Vizualizare lista de probleme: Profesorul poate vizualiza lista de probleme pentru a alege o problema din aceea lista pe care sa o dea ca tema.
6. Adaugare de probleme: Acesta poate sa adauge probleme.

7. Creare de tema: Profesorul poate crea teme alegand o problema din lista de probleme. Acesta poate de asemenea sa seteze un o zi limita pentru incarcarea acesteia.
8. Vizualizarea listei cu elevii care au incarcat temele: Profesorul poate vizualiza lista cu cei care au incarcat temele, apoi cu ajutorul unui modul de cautare acesta poate cauta dupa mai multe criterii cum ar fii titlul temei sau numele elevului.
9. Evaluarea temei: Acesta poate da o recenzie temelor incarcate cat si o nota.

### **3 STUDIUL DE PIAȚĂ / ABORDĂRI EXISTENTE**

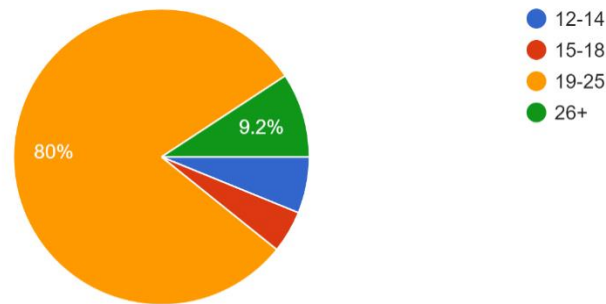
Obiectivul aparitiei unui produs nou pe piata trebuie sa se rezume in totalitate la rezolvarea problemelor potentialilor utilizatori, astfel incat fiecare aplicatie disponibila este orientata unui anumit grup tinta.

Aplicatia pe care o propun se orienteaza catre elevi si profesori, cu scopul facilitarii procesului de invatare si de predare. Consider ca, la momentul actual, in special in Romania, ideea de meditatie este puternic raspandita printre elevi, de multi ori numarul cererilor depasind numarul ofertelor.

De altfel, din cauza contextelor globale din ultimii ani, profesorii au fost nevoiti sa se orienteze catre metode noi de invatare, mai exact platformele de invatare online, considerate poate chiar „nonconformiste” pana in acel moment. Pe parcurs, obisnuindu-se cu noile metode, din ce in ce mai multi profesori au inceput sa utilizeze platformele online de invatare, considerandu-le benefice.

In domeniul educatiei este nevoie tot timpul de metode noi si inovative care sa faciliteze invatarea. Astfel, pentru a realiza o platforma online de invatare suplimentara, utila si fiabila, am creat un chestionar orientat catre nevoile intampinate de elevi, afland de asemenea si parerea unui numar de 65 de persoane despre ideea de „platforma online de invatare”.

Ce vârstă aveți?  
65 responses



Figură 5: Varsta

Mediul de proveniență  
65 responses

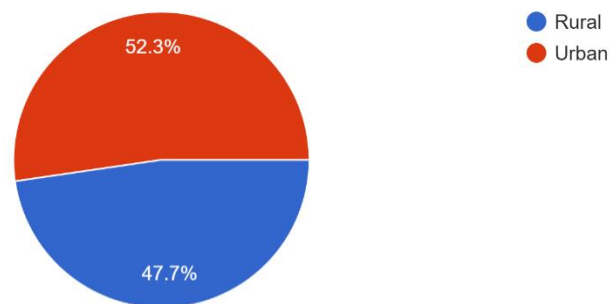
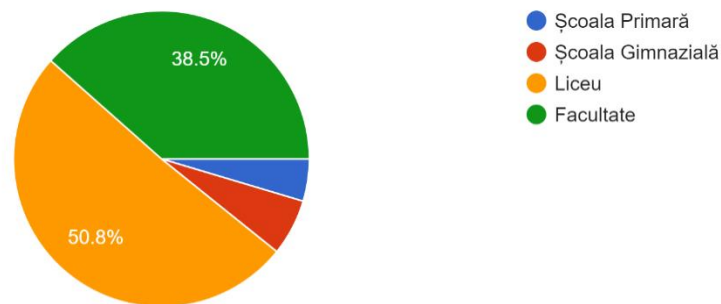


Figura 6: Mediul de provenienta

Cu ajutorul figurilor 1 și 2 putem observa fiabilitatea platformelor online de învățare, relevante și benefice persoanelor de diferite vârste, fiind chestionate 65 de persoane cu vârste cuprinse între 12 și peste 26 de ani, provenite atât din mediul urban (52.3%) cât și din cel rural (47.7%)

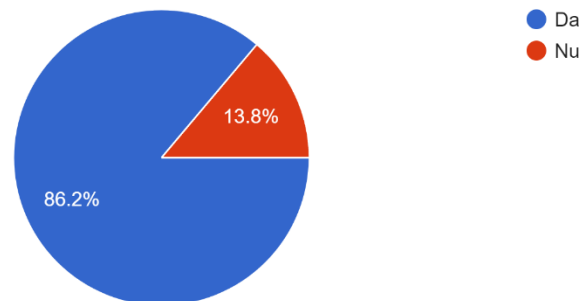


Ultima formă de învățământ finalizată  
65 responses



Figură 7: Ultima forma de invatamant finalizata

În perioada în care ați fost student/elev, ați simțit nevoia de ore suplimentare, în plus față de cele de la școală?  
65 responses



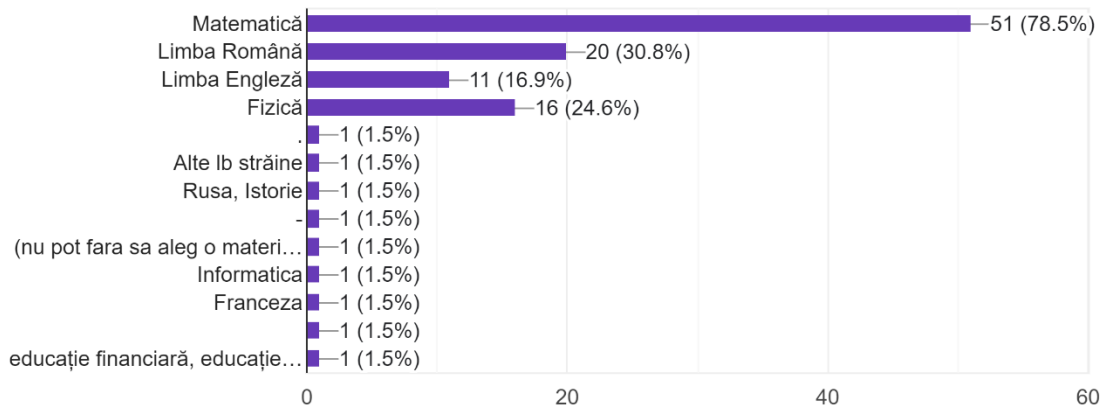
Figură 8

De altfel, cu ajutorul figurilor 3 și 4 observăm faptul că 86,2% dintre persoanele chestionate au simțit nevoia de ore suplimentare în perioada studiilor și doar 13,8% nu au simțit nevoia, persoanele întrebate având ca ultimă formă de învățământ finalizată școala primară, școala gimnazială, liceu și facultate. Astfel, putem spune că nevoia orelor suplimentare apare indiferent de forma de învățământ, indiferent dacă urmează un examen important precum bacalaureatul sau nu.

Este evidențiată nevoia unei schimbări în rândul elevilor și al studenților, pentru facilitarea procesului de învățare al acestora și ajutorarea profesorilor cărora le este greu să ajungă la 100% numărul elevilor care au înțeles și nu mai au nevoie de pregătire și explicații în plus.

Dacă da, la ce materie ați întâmpinat dificultăți?

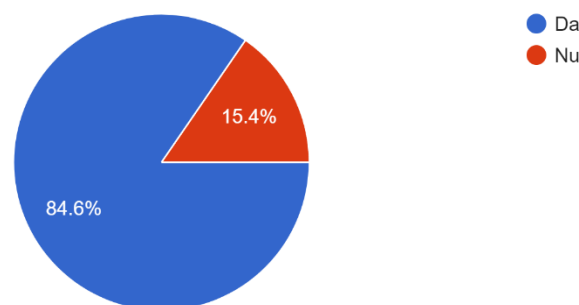
65 responses



Figură 9

Ați apelat la meditații pe parcursul anilor de învățământ din cauza nevoii de suplimentare a orele de la clasă?

65 responses



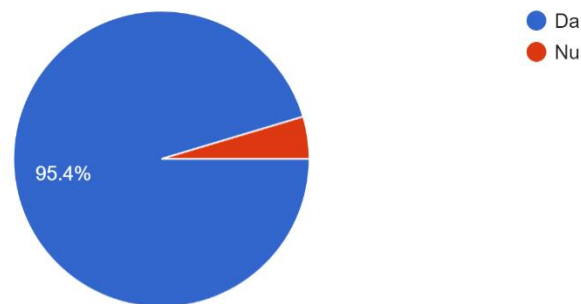
Figură 10

Din figura 5 reiese faptul ca doar 4 persoane din 65 nu au intampinat dificultati in perioada scolii. Consider ca numarul persoanelor ce nu au avut probleme pe parcursul anilor de studii este unul extraordinar de mic si trebuie schimbat. Acest fapt m-a motivat in realizarea aplicatiei de invatare suplimentara si sper sa reusesc sa aduc o imbunatatire si un beneficiu atat elevilor cat si profesorilor.

De asemenea, privind figura 6 observam ca 84,6% dintre persoanele chestionate au apelat la meditatii din cauza nevoi de a suplimenta orele de la clasa si doar 15,4% nu au facut meditatii. Procentul persoanelor care au facut meditatii este unul foarte mare, de altfel si nevoia si de pregatire suplimentara este uriasa. Sustin faptul ca platforma de invatare suplimentara poate sprijini nevoile elevilor.

Sunteți de acord că meditațiile NU sunt la îndemâna oricui?

65 responses



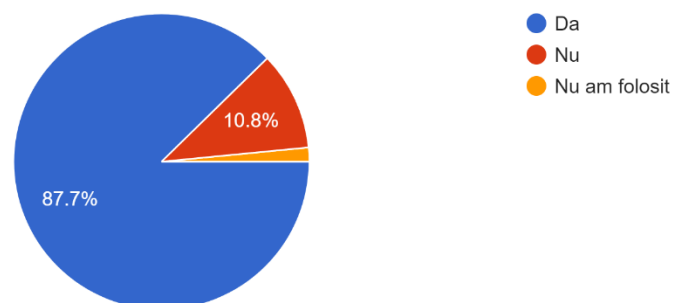
Figură 11

Figura 7 reprezintă un adevăr foarte trist cu care, din păcate, multe persoane se confruntă. Meditațiile din nefericire nu sunt la îndemâna oricui, de altfel mulți elevi chiar dacă au nevoie de pregătire suplimentară nu pot avea parte de ea iar acest lucru îi poate împiedica în dezvoltarea lor.

Sunt de părere că toată lumea ar trebui să aibă acces la pregătire suplimentară în caz de nevoie și consider că platforma prezentată poate ajuta prin scăderea numărului elevilor care nu au acces la pregătire suplimentară și a celor care apelează la meditații.

Considerați că platformele de învățare online (ex Google Classroom, Moodle etc.) sunt benefice atât pentru profesori cât și pentru elevi/studenți?

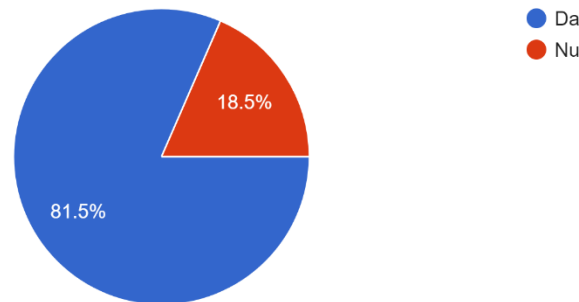
65 responses



Figură 12

Considerați că o platformă online de învățare suplimentară, nouă și ușor de folosit, poate reduce numărul elevilor și al studenților care apelează la meditații ?

65 responses



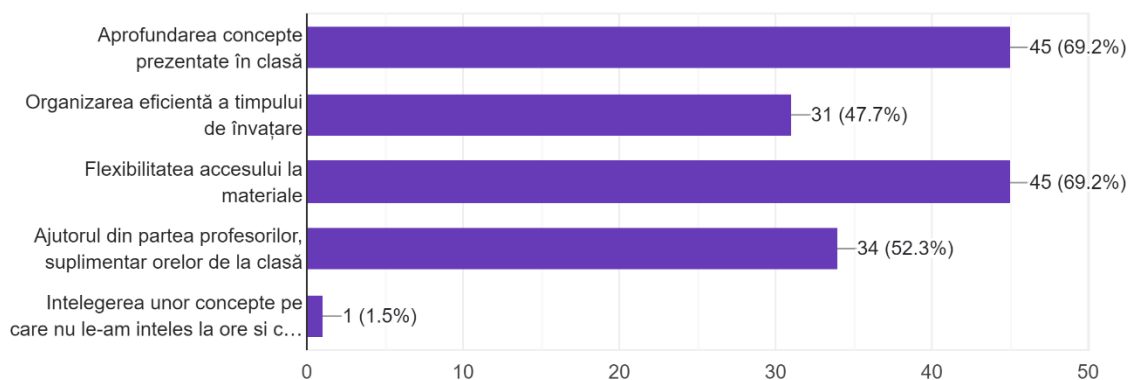
Figură 13

Analizand figurile 8 si 9 vedem parerea persoanelor chestionate referitor la platformele online si beneficiile lor. In figura 8 se observa ca doar 10,8% nu considera benefice aplicatiile de invatare online deja existente pe piata si 1,5% nu au folosit, ramanand un procentaj foarte mare de persoane ce considera benefice aplicatiile de educatie online.

De asemenea, in figura 9 este precizat faptul ca 81,5% dintre persoane sunt de parere ca o noua platforma online de invatare suplimentara poate reduce numarul elevilor care apeleaza la meditatii intarind ideea pe care o sustin si obiectivul pe care mi l-am propus in realizarea aplicatiei prezentate.

Care sunt motivele pentru care ați folosi o platformă online de învățare suplimentară?

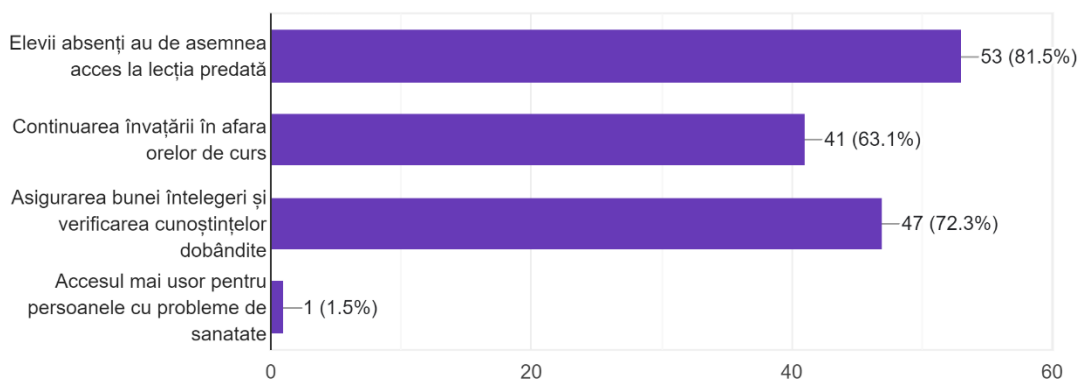
65 responses



Figură 14

Care considerați că ar fi beneficiile combinării platformei online de învățare suplimentară cu metoda tradițională (orele de la școală)?

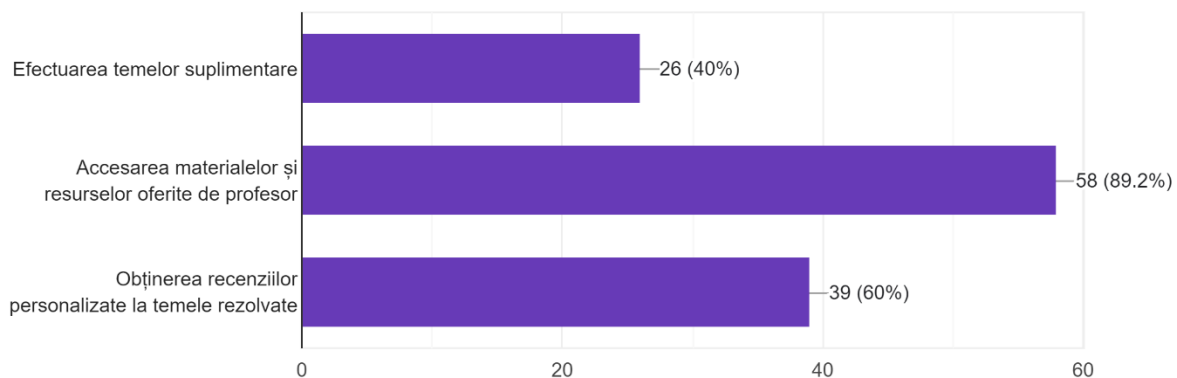
65 responses



Figură 15

Care considerați că ar fi beneficiile studenților folosirii platformei online de învățare suplimentară?

65 responses



Figură 16

În prezent, în multe școli și instituții de învățământ, profesorii folosesc platformele de învățare online pentru trimiterea temelor sau, mai rar, pentru postarea cursurilor din ziua respectivă. Însă, platforma pe care eu o propun vizează în special învățarea suplimentară și înlocuirea pe cât posibil a meditațiilor prin activități, cursuri și explicații suplimentare, ci nu strict pentru trimiterea temelor sau a lecției de zi.

Consider ca o platforma online noua si usor de folosit poate fi benefica invatamantului si dezvoltarii persoanelor.

Observam in figura 10 ca niciuna dintre persoanele chestionate nu a mentionat faptul ca nu ar dori sa foloseasca o asemenea persoana, in plus raspunsurile primite sunt pro unei asemenea platforme deoarece nevoia invatarii suplimentare este mare.

De asemenea, figurile 11 si 12 se refera la beneficiile platformei, in perseptiva elevilor dar si a profesorilor. In plus, niciuna dintre persoanele chestionate nu a mentionat vreun dezavantaj in folosirea acestei noi platforme.

Pentru a exemplifica, analizam figura 11 care se refera la imaginea de ansablu si beneficiile pe care platforma le poate avea in invatamant, prin combinarea sa cu metoda de invatare traditionala, cea din salile de clasa. Persoanele chestionate considera un avantaj combinarea celor 2 metode de invatare, in vederea unei mai bune intelegeri a cursurilor, continuarea invatarii si comunicarea cu profesorii in afara programului, in plus, chiar sugestia anonima privind accesul persoanelor cu probleme de sanatate ofera o imagine de ansamblu asupra castigului utilizarii platformei. Se observa faptul ca verificarea cunostintelor dobandite are o mare importanta si este pe locul 2 in clasament, la mica diferenta de locul 1 care vizeaza accesul elevilor absenti la materialele si cursurile predate in ziua respectiva.

Ultima intrebare a chestionarului (figura 12) ii vizeaza strict pe elevi si avantajele pe care acestia le-ar avea utilizand platforma prezentata. De asemenea, fiecare persoana a ales drept raspuns minim 2 variante, rezultand beneficiile si suportul pe care cursantii le-ar avea prin folosirea platformei online de invatare suplimentara.

Consider ca, raspunsurile primite in acest chestionar imi intaresc si imi sprijina ideea de necesitatea a unei asemenea platforme, ce poate fi la indemana cat mai multor persoane si poate aduce ajutor invatamantului actual si elevilor ce se confrunta cu diferite probleme privind educatia, prin fiabilitatea sa, usurina folosirii din orice tip de profil si optiunile pe care aceasta le ofera profesorilor si elevilor.

## 4 SOLUȚIA PROPUȘĂ

Abordarea pe care am ales să merg provine din documentarea pieței la momentul actual. Astfel după o analiză amănunțită am observat că o bună parte din marile companii precum JPMorgan Chase Bank, Intellibus, Deutsche Bank și multe altele folosesc Java Spring Boot pentru backend. Pentru frontend am ales să merg pe Angular folosind typescript cu toate că React este mai popular în rândul aplicațiilor mici dar pentru proiectele mari este mai folosit Angular. Pentru baza de date am ales să merg pe o bază de date relațională, respectiv MySQL, aceasta fiind open-source, de asemenea fiind cunoscută pentru performanță și scalabilitate.

### 4.1 Spring Boot

Spring este cel mai popular framework de Java. Titlatura de cel mai popular framework este obținută din viteza, simplitatea și productivitatea acestuia.

Spring este un framework pentru a construi aplicații Java, open-source, care oferă o infrastructură complexă și ușor de menținut pentru dezvoltarea aplicațiilor. Principalele avantaje în folosirea Spring:

1. Inversion of control – acesta este un principiu de proiectare care oferă modularitate programului, oferind frameworkului control asupra unor porțiuni de cod sau a unor obiecte. Acest lucru este realizabil prin Dependency Injection. Dependency Injection este mecanismul prin care controlul este inversat de la programator la framework.
2. Accesul la date – simplificarea accesului la date folosind tehnologii precum Hibernate, JPA (Java persistence API), MyBatis, EclipseLink. Acestea reduc complexitatea interacțiunii cu baza de date.
3. Ideal pentru construirea de RESTful API datorită anotărilor simpliste: @RestController, @GetMapping, @PostMapping simplifică definiția și gestionarea. Aceste anotări fac codul ușor de citit și întreținut permițând o dezvoltare ușor facilă a endpointurilor de tip CRUD (create, read, update, delete).
4. Securitatea – securitatea este ușor de facilitată datorită frameworkului Spring Security JWT (json-web token) care ajută în autentificare și autorizare. De asemenea oferă protecție împotriva atacurilor de tip CSRF

### 4.2 Angular

Angular este un framework open-source bazat pe TypeScript, care rulează pe Node.js și este utilizat pentru crearea aplicațiilor web.

De ce TypeScript și nu JavaScript?

Unul dintre cele mai mari avantaje pe care consider că le are TypeScript asupra JavaScript este pentru că acesta are erori de compilare datorită tipizării statice în loc de tipizarea dinamică în JavaScript.

Caracteristici ale Angular

Angular este bazat pe componente. Componentele ofera utilizatorului abilitatea de a defini propriile componenete personalizate. Aceste pot fi folosite simultan cu elementele HTML standard, precum `<div></div>`. Un exemplu de folosire al acestei caracteristici este crearea unei componente `<add-problem></add-problem>` care este o pagina cu un formular pentru adaugare a exercitiilor. Aceste poate fi folosit ca un „obiect” in orice alta componenta a aplicatiei. Astfel acest stil de programare este similar cu programarea orientata pe obiecte, aducand toate beneficiile ale acestuia cum ar fi modularitatea, reutilizarea, intretinerea si posibilitatea de extindere.

Printre beneficiile majore ale utilizării componentelor se numără:

- Organizare și Claritate: Codul este mult mai bine organizat și clar, facilitând înțelegerea și navigarea.
- Reutilizarea Codului: Componentele pot fi reutilizate în diferite părți ale aplicației, reducând redundanța și economisind timp.
- Întreținere Facilă: Codul modular și componentizat este mai ușor de întreținut și actualizat, permițând implementarea mai rapidă a modificărilor și a noilor funcționalități.

### 4.3 MySQL

MySQL este un sistem de gestionare a bazelor de date relaționale (RDBMS) open-source. Este unul dintre cele mai populare și utilizate motoare de baze de date din lume, cunoscut pentru performanța sa, fiabilitatea și ușurința în utilizare. MySQL este dezvoltat, distribuit și susținut de Oracle.

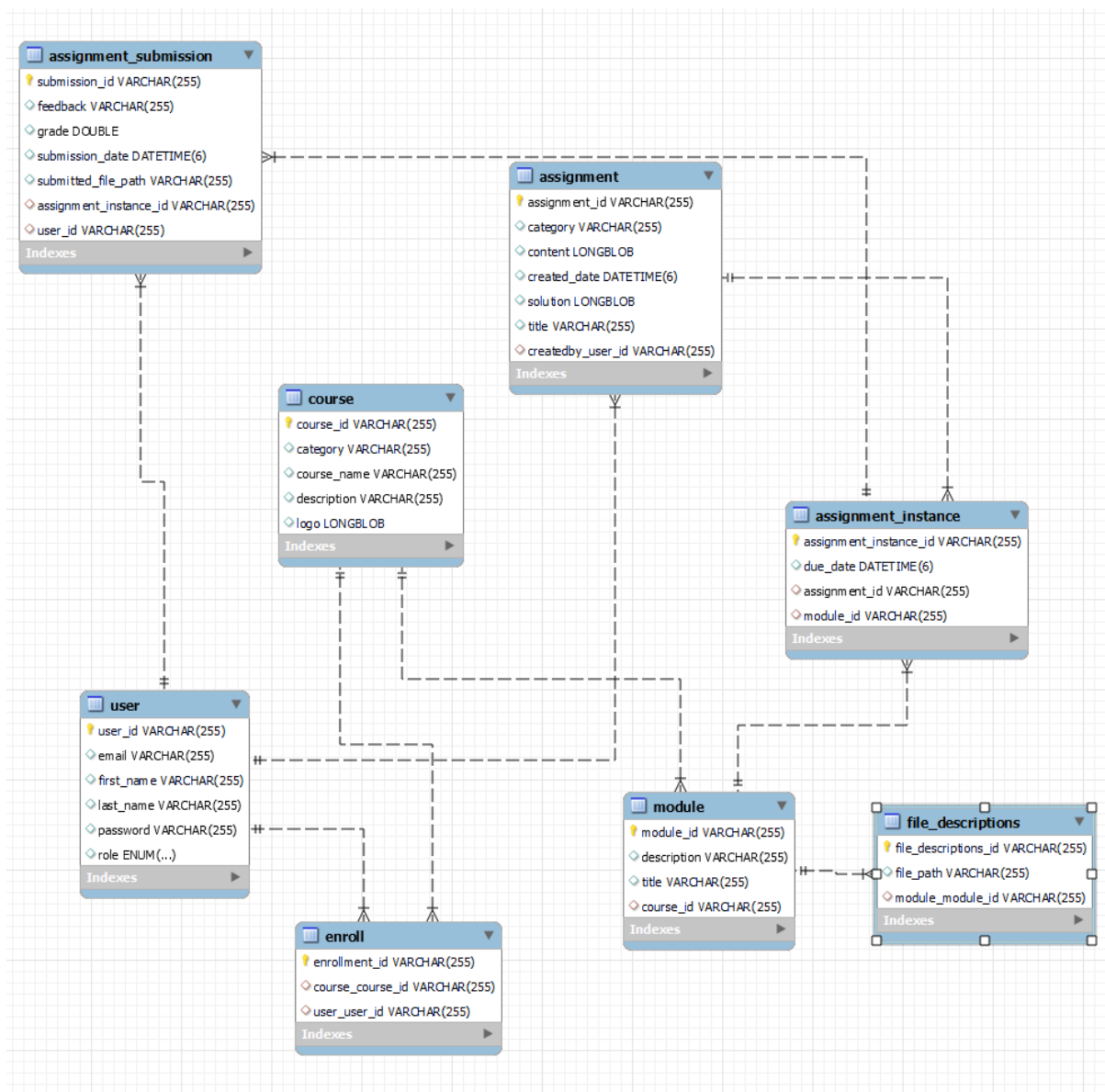
### 4.4 Open-Source

Am menționat faptul ca alegerile mele in construirea aplicatiei sunt open-source. Astfel voi argumenta care sunt avantajele:

1. Transparența: dezvoltatorii pot inspecta codul sursa pentru a observa daca acesta este functional, de asemenea acesta este un factor bun si din punct de vedere al securitatii.
2. Securitatea: datorita transparenței dezvoltatorii pot analiza codul de vulnerabilitati ascunse. Legea lui Linus Torvalds „Given enough eyeballs, all bugs are shallow” – aceasta inseamna cu cat mai multa lume lucreaza si testeaza aplicatia problemele aplicatiei dispar.
3. Valoare educationala
4. Suport si comunitate: datorita faptului ca este open-source suportul si comunitatea este pe masura fiind foarte usor sa gasesti informatii. De asemenea toti dezvoltatorii pot aduce contributii.
5. Inovare si dezvoltare rapida: datorita faptului ca toata lumea poate contribuii duce la dezvoltarea rapida si aducerea a noi inovatii in tehnologie.



## 4.5 Arhitectura bazei de date



Figură 17: Structura bazei de date

1. Tabela user: Aceasta contine informatiile despre utilizatori. Coloanele sunt: user\_id cheie primara, email first\_name last\_name, password si role.
2. Tabela course: Aceasta contine informatiile despre cursuri. Coloanele sunt course\_id cheie primara, category, course\_name, description si logo.
3. Tabela module: Aceasta contine informatii despre modulele din curs, fiecare modul fiind legat de un curs. Coloanele sunt module\_id cheie primara, description, title si course\_id cheie straina.
4. Tabela file\_description: Aceasta contine caile pentru fisierele din module. Coloanele sunt: file\_descriptions\_id cheie primara, file\_path, module\_module\_id cheie straina.

5. Tabela assignment: Aceasta contine informatii despre exercitiile care pot fi folosite ca teme. Coloane: assignment\_id cheie primara, category, content, created\_date, solution, title si createdby\_user\_id cheie straina
6. Tabela assignment\_instance: Aceasta tabela contine informatii despre teme, fiind legata de module. Coloane: assignment\_instance\_id cheie primara, due\_date, assignment\_id cheie straina, module\_id cheie straina
7. Tabela assignment\_submission: Aceasta tabela contine informatii despre temele incarcate ale studentilor. Coloane: submission\_id cheie primara, feedback, grade, submission\_date, submitted\_file\_path, assignment\_instance\_id cheie straina si user\_id cheie straina.
8. Tabela enroll: In aceasta tabela este retinut ce utilizator este inregistrat la un anumit curs. Coloane enrollment\_id cheie primara, course\_course\_id cheie straina si user\_user\_id cheie straina.

## 4.6 Docker

Docker este o platformă open-source care automatizează deployment-ul aplicațiilor, permițând rularea acestora într-un mediu izolat. Docker facilitează împachetarea unei aplicații într-un singur container, ceea ce contribuie la principiile de dezvoltare continuă și livrare continuă.

În aplicație vor fi trei servicii: frontend, backend și baza de date, orchestrate printr-un fișier docker-compose.yml. Fiecare serviciu rulează într-un container propriu și este interconectat printr-o rețea comună. Astfel, vor rula trei containere:

1. mysqldb: Containerul care rulează MySQL.
2. Spring Boot RESTful Web Services: Containerul care rulează backend-ul aplicației, configurat pentru a se conecta la baza de date și la frontend.
3. frontend: Containerul care rulează serverul Angular pentru partea de frontend.

De asemenea, utilizez o rețea comună, springboot-mysql-net, pentru a permite containerelor să comunice eficient între ele.

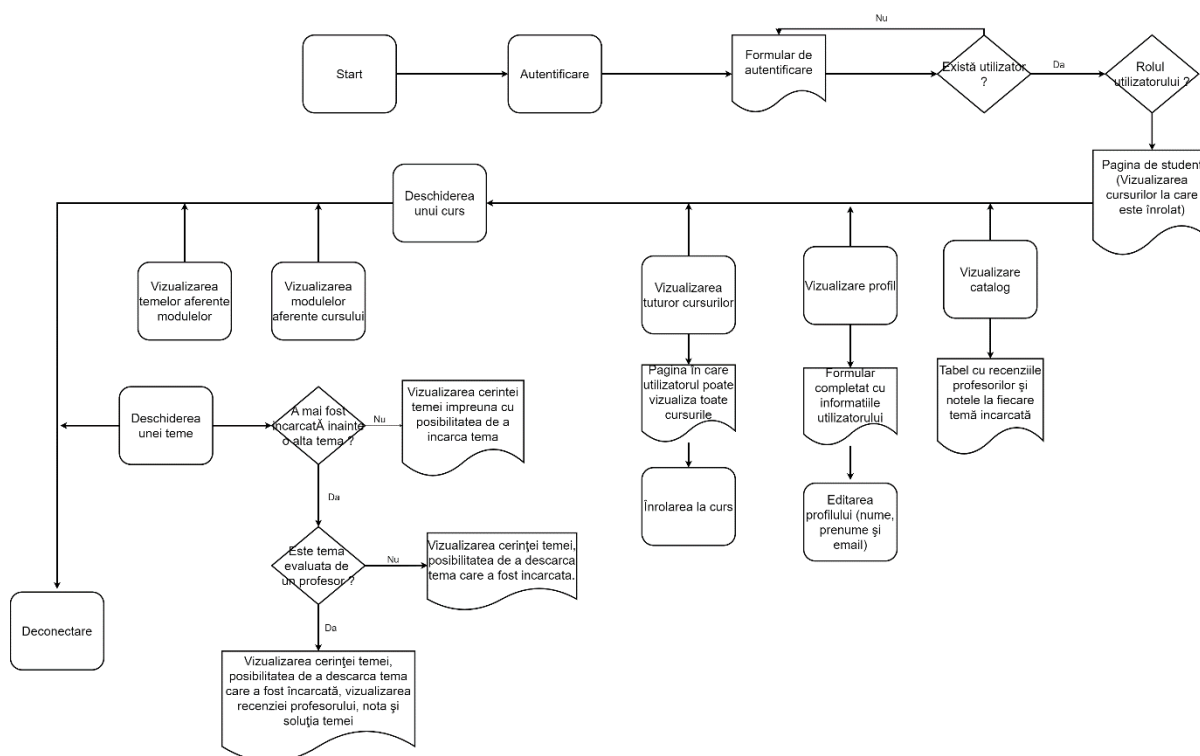
## 4.7 Principiul dezvoltare continua si livrare continua

Obiectivul principiului dezvoltare continua si livrarea continua este de a crește rapiditatea cu care este dezvoltată o aplicație și de a îmbunătăți calitatea produsului finit. Principalul atu al acestui principiu este faptul ca îmbunătățește experiența dezvoltatorului dar și a consumatorului aplicației, aplicația fiind livrată către acesta în cicluri stabilite de timp permițând o mai bună comunicare între aceștia. Consumatorul poate propune schimbări în aplicație la orice moment de timp, iar dezvoltatorul poate implementa aceasta schimbare și o poate livra într-o nouă iterație.

Astfel acest principiu se pliază cel mai bine pe un mod de dezvoltare iterativ.

## 4.8 Descrierea fluxului funcționalităților

Am ales să folosesc o diagrama simplă pentru a evidenția punctele critice de decizie și pentru a ilustra fluxul funcționalităților. Aceste diagrame sunt vitale pentru înțelegerea modului în care diferitele părți ale sistemului lucrează împreună pentru a îndeplini anumite funcționalități.



Figură 18: Diagrama de flux pentru utilizatorii cu rolul de student

În această diagramă sunt evidențiate toate funcționalitățile pe care le poate efectua un utilizator în rolul de student. La autentificare dacă credențialele sunt corecte acesta este redirectionat pe o pagină pe care poate vizualiza toate cursurile la care este înrolat. Descrierea fluxului funcționalităților din pagina inițială:

- Vizualizarea catalogului cu posibilitatea de a căuta recenziile profesorilor la un anumit curs
- Vizualizarea profilului cu posibilitatea de a modifica elemente ale profilului cum ar fi nume, prenume și email
- Vizualizarea tuturor cursurilor cu posibilitatea de a se înrola la orice curs care apare

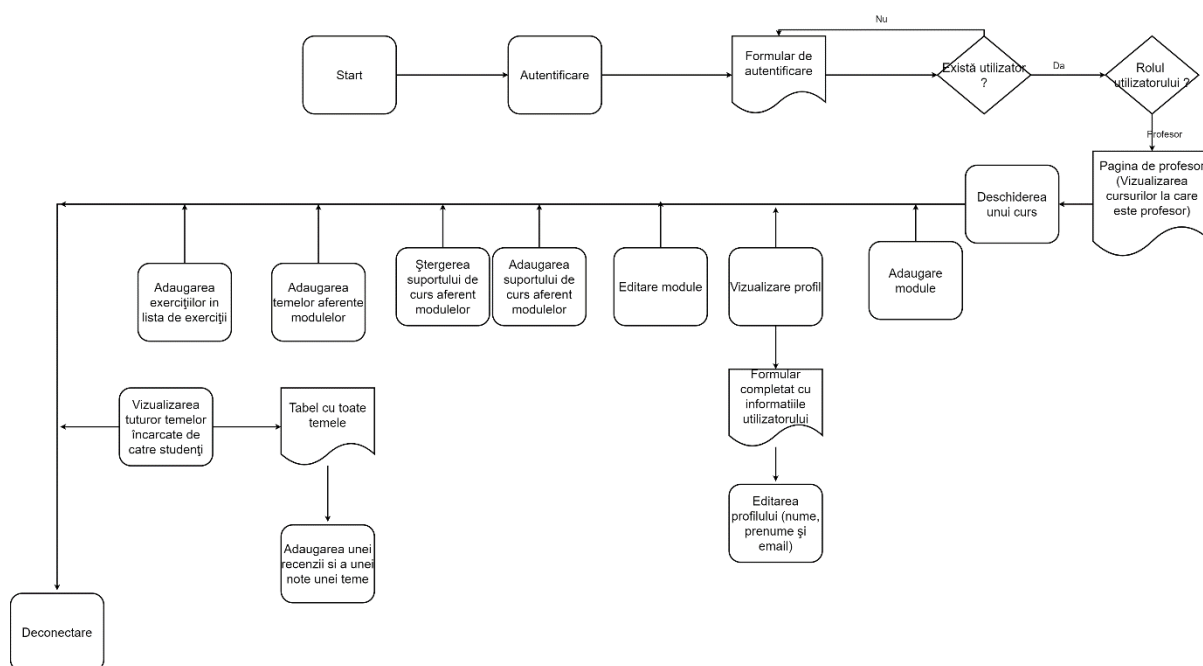
Descrierea fluxului funcționalităților după deschiderea unui curs la care studentul este înrolat:

- Vizualizarea modulelor de curs

- Vizualizarea temelor

La deschiderea unei teme:

- Dacă o tema nu este încărcată acesta vede cerința și are posibilitatea de a încărca.
- Dacă este deja o tema încărcată de către care încă nu este evaluată de un profesor studentul poate vedea cerința temei, ultima încărcare, posibilitatea de a descărca ultima încărcare și posibilitatea de a încărca pentru a modifica ultima încărcare.
- Dacă tema este evaluată de un profesor studentul poate observa recenziile profesorului, nota cât și posibilitatea de a vizualiza cerința, încărca o altă temă și descărca ultima încărcare.

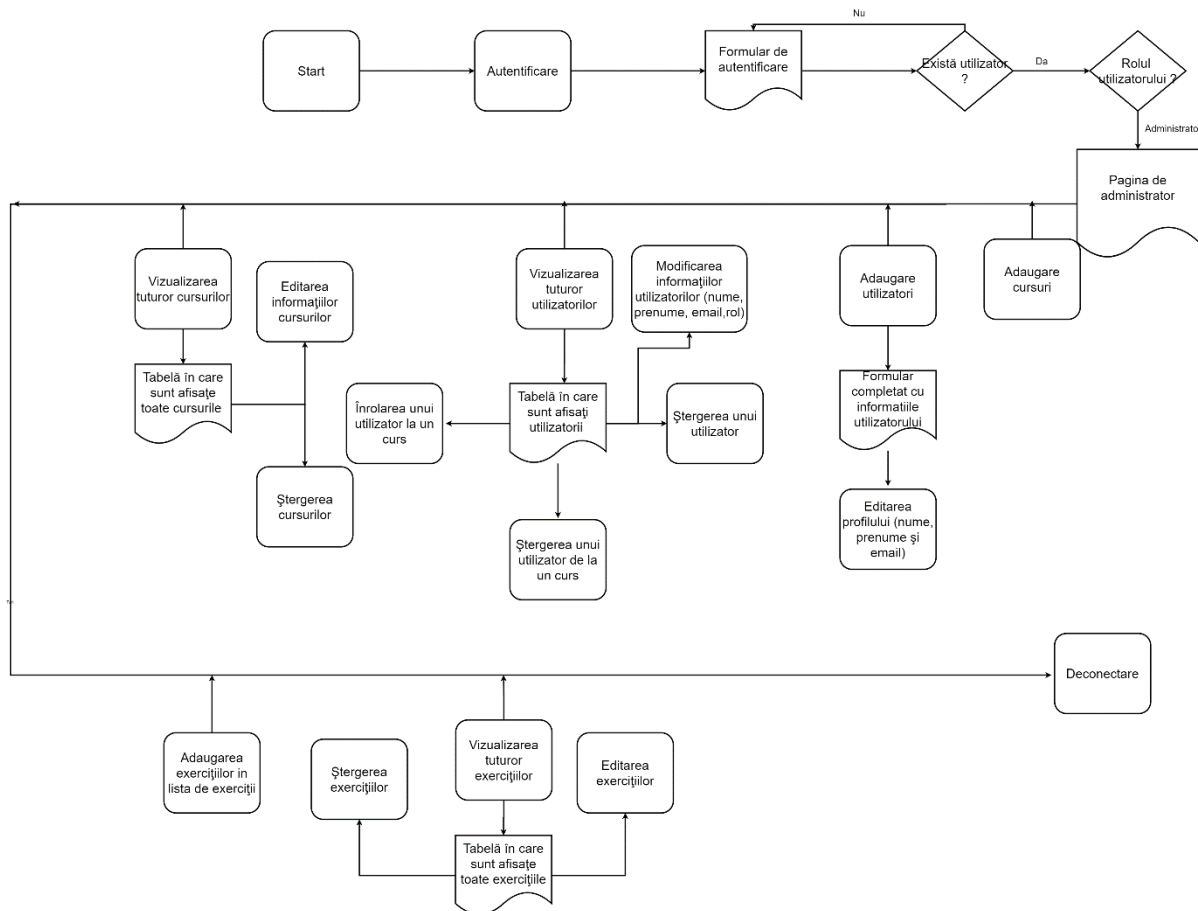


Figură 19: Diagrama de flux pentru utilizatorii cu rolul de profesor

În această diagramă sunt evidențiate toate funcționalitățile pe care le poate efectua un utilizator în rolul de profesor. Dacă profesorul deschide un curs poate realiza următoarele acțiuni:

- Adăuga un modul
- Editarea unui modul
- Ștergerea unui modul
- Adăugarea suportului de curs aferent modulelor
- Ștergerea suportului de curs aferent modulelor
- Adăugarea temelor aferente modulelor
- Adăugarea exercițiilor în lista de exerciții

- Vizualizarea listei de exercitii, modificarea si stergerea exercitiilor din lista la care pentru o mai usoara folosinta exista un sistem de cautare specializat
- Vizualizarea tuturor temelor incarcate de catre studenti unde apare un tabel cu toate temele incarcate pentru cursul respectiv la care profesorul poate cauta daca doreste sa corecteze temele cu un anumit titlu



Figură 20: Diagrama de flux pentru utilizatorii cu rolul de administrator

În această diagramă sunt evidențiate toate funcționalitățile pe care le poate efectua un utilizator în rolul de administrator.

- Adăugarea cursurilor
- Adăugarea utilizatorilor
- Vizualizarea tuturor utilizatorilor de unde administratorul poate edita informațiile utilizatorului (nume, prenume, email și rol), înrola profesori și utilizatori la anumite cursuri, șterge utilizatori de la anumite cursuri și șterge utilizatori
- Vizualizarea tuturor cursurilor de unde administratorul poate edita informațiile cursului și șterge cursul
- Vizualizarea listei de exercitii, modificarea și stergerea exercitiilor din lista la care pentru o mai ușoară folosință există un sistem de căutare specializat
- Adăugarea exercitiilor în lista de exercitii

## 5 DETALII DE IMPLEMENTARE

În acest capitol vom explora în detaliu intrând amanuntit în provocările întâmpinate pe parcursul dezvoltării aplicației. Pentru a intra mai în amanunt și a oferi o perspectivă clară asupra problemelor, vom folosi prezentarea secvențelor de cod esențiale, pseudo-cod, diagrame de secvență și de resurse găsite în timpul cercetărilor. Resursele prezintă diferite perspective incluzând articole, studii de caz și documentații tehnice folosite pentru a privi din perspective diferite soluțiile problemei.

În plus, capitolul va detalia modul în care studiul de caz și tehnologiile propuse au fost de ajutor în dezvoltarea soluției finale, accentuând modul în care ele au contribuit la rezolvarea problemelor. De asemenea, vom analiza strategiile de arhitectură abordate la diferite impedimente în structura aplicației.

Prin prezentarea detaliată a algoritmilor și structurilor utilizate, împreună cu etapele de dezvoltare și corectitudinea soluțiilor implementate, acest capitol va oferi o imagine completă și transparentă a procesului de dezvoltare tehnică.

### 5.1 Autentificare și securitate

Pentru a înțelege metoda pe care am ales-o în autentificarea și autorizarea utilizatorilor în aplicație, cât și elemente ce țin de securitate, voi prezenta mai multe opțiuni.

#### 5.1.1 Autentificare pe baza cookie-urilor

Autentificarea pe baza cookie-urilor funcționează în următoarea manieră:

1. Utilizatorul completează email-ul și parola și trimite o cerere către server de tip post.
2. Serverul verifică validitatea credențialelor. Dacă acestea sunt corecte, creează o sesiune pe care o reține și trimite id-ul sesiunii alături de informațiile aferente într-un cookie către client.
3. La fiecare cerere ulterioară, clientul trimite automat informațiile din cookie. Astfel, serverul poate verifica dacă utilizatorul este autorizat sau nu.

Avantajele Autentificării pe bază de Cookie-uri

- Gestionare Automată: Cookie-urile sunt administrate automat de browser, simplificând astfel codul pe partea de client.
- Validare și Invalidare Facilă: Deoarece serverul menține starea, acesta poate valida sau invalida sesiunile cu ușurință.

Dezavantajele Autentificării pe bază de Cookie-uri

- Scalabilitate: Serverul trebuie să păstreze starea sesiunilor, ceea ce face dificilă scalarea verticală.
- Blocaje la trafic mare: Administrarea sesiunilor pe server poate duce la blocaje în cazul unui trafic mare de cereri.

### 5.1.2 OAuth 2.0

OAuth 2.0 este un protocol de autorizare care permite aplicațiilor terțe să obțină acces limitat la resursele utilizatorilor fără a necesita trimiterea de credențiale.

Pasii prin care se realizeaza autorizarea sunt urmatoarii:

1. Clientul este redirectionat către o aplicație terță aleasă, unde utilizatorul introduce credențialele, dacă nu este deja autorizat pe acea aplicație.
2. Aplicatia terța autorizeaza autentificarea și utilizatorul este redirectionat către aplicatia principala împreună cu un cod de autorizare.
3. Clientul creează o cerere de tip POST ce conține codul de autorizare de la aplicatia terța pentru a obține un token de acces în aplicatie.
4. La fiecare cerere ulterioară tokenul de acces este trimis în antetul cererii, astfel se face autorizarea.

Avantajele autentificării pe bază de OAuth 2.0

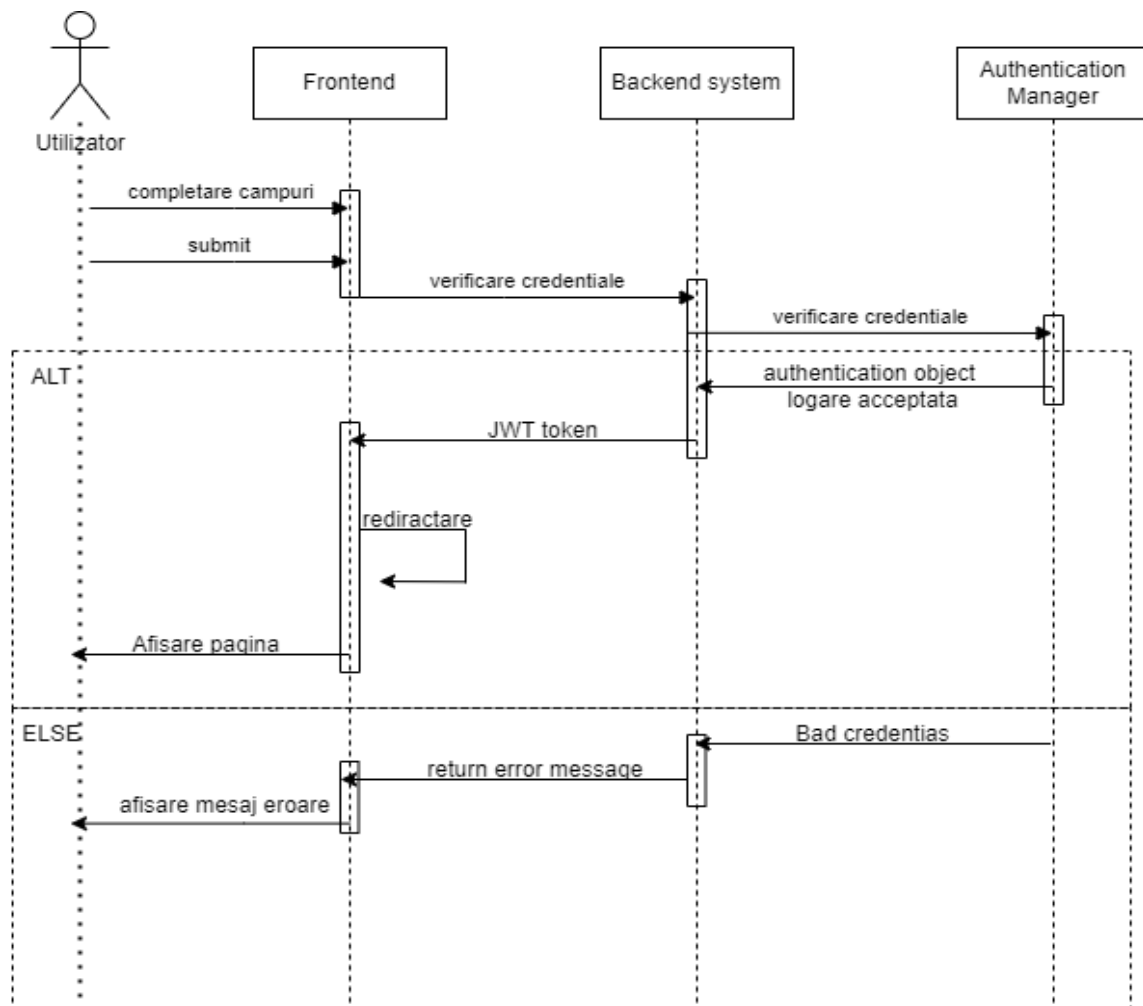
- Autentificarea numai este strâns legată de credențialele utilizatorului
- Suportă diverse tipuri de fluxuri de autorizare și de tokeni cum ar fi opaque tokens, SAML assertions, JWT tokens.

Dezavantajele Autentificării pe bază de Cookie-uri

- Cea mai mare vulnerabilitate pe care o poate avea acest tip de autorizare este că codul de autorizare din pasul 2 să fie interceptat de un atacator.
- Se bazează pe o aplicație externă pentru a gestiona autorizarea.

### 5.1.3 JWT token

Autentificarea pe baza JSON Web Token este o metodă fără stare ce permite transmiterea de informații între server și client într-un mod securizat prin intermediul unui token semnat. Pentru a descrie pașii de autentificare și autorizare folosind JSON Web Token, voi utiliza o diagramă de secvență (FIGURA X).



Figură 21: Diagrama de secventa autentificare

Astfel utilizatorul completează câmpurile (email si parola) și apasă autentificare. Clientul trimite o cerere de tip POST ce conține credențialele utilizatorului către server. Serverul utilizează un Authentication Manager care facilitează verificarea credențialelor in baza de date. Dacă credențialele sunt corecte, acesta returnează un obiect de autentificare, iar clientul primește un jwt token criptat ce conține proprietățile utilizatorului (email, rol, data expirarii tokenului). Clientul îl salvează în memoria locala a motorului de căutare si îl trimite ulterior cu fiecare cerere pe care clientul o face. Dacă credențialele sunt incorecte Authentication managerul întoarce HTTP STATUS BAD CREDENTIALS, iar utilizatorului ii este afișat un mesaj de eroare.

Avantajele autentificării pe bază de OAuth 2.0

- Scalabilitate: Datorită faptului că este statless, nu necesită menținerea informațiilor de către server, ceea ce face aplicatia scalabila.
- Performanta: Tokenul poate conține informații (claims), eliminând astfel necesitatea unor interogări frecvente în baza de date, cum este cazul autentificării folosind sesiuni cookie.



- Interoperabilitate: Poate fi utilizat cu orice limbaj de programare, iar datorită faptului că este statless se pliază foarte bine pe principiile RESTful
- Securitate: Token-urile sunt semnate digital asigurând integritatea datelor

#### Dezavantajele Autentificării pe bază de Cookie-uri

- Revocarea token-ului înainte de expirare este complicată. Soluțiile includ utilizarea unei liste de revocare sau emiterea de token-uri cu durată scurtă de viață.
- Securitate: Deoarece payload-ul (claims) JWT nu este criptat, ci doar semnat, informațiile sensibile nu ar trebui incluse în token.

Rezultatul analizei mele au fost ca utilizarea JSON Web Token este cea mai buna alegere pentru această aplicație.

```
no usages
@Bean
public SecurityFilterChain securityFilterChain(HttpSecurity httpSecurity) throws Exception {
    httpSecurity
        .csrf(AbstractHttpConfigurer::disable)
        .cors(httpSecurityCorsConfigurer -> httpSecurityCorsConfigurer.configurationSource(corsConfigurationSource()))
        .authorizeHttpRequests(auth -> auth
            .requestMatchers(...patterns: "/api/auth/**").permitAll()
            .requestMatchers(...patterns: "/api/admin/**").hasRole("ADMIN")
            .anyRequest().authenticated())
        .sessionManagement(sess -> sess.sessionCreationPolicy(SessionCreationPolicy.STATELESS))
        .authenticationProvider(authenticationProvider)
        .addFilterBefore(jwtAuthFilter, UsernamePasswordAuthenticationFilter.class);

    return httpSecurity.build();
}
1 usage
```

Figură 22: Configurarea securitatii

Principala configurație a securității se află în imaginea din figura x. Astfel metodă `securityFilterChain` configurează CORS, definește regulile de autorizare și adaugă filtrul de autentificare JWT înainte de `UsernamePasswordAuthenticationFilter`. Acest filtru este o clasă `JwtAuthenticationFilter` care extinde `OncePerRequestFilter` ce asigura faptul că filtrul este apelat o singură dată pe cerere și suprascrie metodă `doFilterInternal` care verifică antetul „Authorization” pentru a extrage și valida tokenul JWT.

```

1 usage
public ResponseEntity<AuthenticationResponse> login(AuthenticationRequest request) {
    try {
        authenticationManager.authenticate(
            new UsernamePasswordAuthenticationToken(
                request.getEmail(),
                request.getPassword()
            )
        );
    } catch (BadCredentialsException badCredentialsException) {
        return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body(
            AuthenticationResponse.builder()
                .message("Authentication failed: email or password are wrong")
                .build()
        );
    }
    User user = userRepository.findByEmail(request.getEmail());
    Map<String, Object> claims = new HashMap<>();
    claims.put("role", user.getRole());
    String jwtToken = jwtService.generateToken(claims, user);
    return ResponseEntity.ok(AuthenticationResponse
        .builder()
        .token(jwtToken)
        .build());
}

```

Figură 23: Autentificarea

În figura x este prezentată metodă în care ajunge cererea de la client când utilizatorul încearcă să se autentifice. După verificarea credențialelor dacă sunt valide este trimis un token ce conține rolul și alte detalii cum ar fi data expirării către client.

În serverul de Angular salvez tokenul în baza de date locală a browserului și am creat un interceptor prin care trece fiecare request și adaugă în antetul cererii tokenul.

```

5+ usages
export const authInterceptor: HttpInterceptorFn = (req : HttpRequest<unknown>, next : HttpHandlerFn) => {
    const router = inject(Router);
    const token = localStorage.getItem( key: 'studdyBuddy_token');
    if (token) {
        let decodedToken = jwtDecode(token);
        const isExpired = decodedToken && decodedToken.exp ? decodedToken.exp < Date.now() / 1000 : false;
        if (isExpired) {
            localStorage.removeItem( key: 'studdyBuddy_token');
            router.navigate( commands: ['/login']);
            throw new Error('Session expired. Redirecting to login.');
```

Figură 24: Interceptorul pentru adaugarea antetului ce conține tokenul JWT

Figura x reprezintă codul sursă pentru adăugarea antetului ce conține tokenul JWT la fiecare cerere pe care o face către server. Tokenul este luat din local storage-ul browserului, se face verificarea dacă token-ul este expirat utilizatorul este trimis către pagină de login și îi apare mesajul de eroare aferent, iar dacă nu este expirat cererea este clonată și adăugat token-ul.

## 5.2 Accesul la date

O alta problema pe care am întâmpinat-o este găsirea celui mai optim mod pentru aplicația mea în legătura cu accesul la date. În Spring Boot am găsit mai multe opțiuni pe care le voi prezenta și voi justifica alegerea făcută.

### 5.2.1 JDBC Template

JDBC este o interfață care definește cum clienții pot accesa baza de date. Această metodă de interacțiune cu baze de date este orientată către bazele de date relationale și oferă metode pentru a face cereri, actualizări, cereri și apeluri de proceduri în baza de date.

JdbcTemplate este o clasă centrală în pachetul JDBC care simplifică utilizarea acestuia și ajută la evitarea erorilor comune cum ar fi gestionarea excepțiilor.

Avantaje:

- Mai performant pentru operațiuni de bază.
- Suportă cereri și proceduri stocate

Dezavantaje:

- Scrierea a mult cod până la executarea unei cereri cum ar fi, creare conexiunii, crearea cereri, închiderea conexiunii.
- Repetarea codului de la o bază de date la alta implică timp.

### 5.2.2 MyBatis

MyBatis este un framework de persistență care suportă SQL personalizat. Acesta oferă suport pentru mapearea SQL și XML la obiecte Java.

Avantaje:

- Flexibil pentru cereri complexe.
- Este mai flexibil când vine vorba de arhitectura bazei de date.

Dezavantaje:

- MyBatis solicită mapearea manuală a câmpurilor obiectelor la rezultatele cererilor SQL.
- Mai puțin orientat pe obiecte.

### 5.2.3 Spring Data JPA

Spring Data JPA este un modul al framework-ului Spring ce simplifică accesul la baze de date folosind API-ul Java Persistence (JPA). JPA este o specificație Java care descrie o metodologie de mapeare a datelor între un model de obiecte și o bază de date relațională. Hibernate este

una dintre cele mai populare implementări ale JPA și oferă un set complet de funcționalități pentru managementul persistenței datelor.

Avantaje:

- Simplificare și Reducerea Codului: Elimină necesitatea de a scrie manual codul de acces la date, reducând astfel complexitatea și cantitatea de cod necesar.
- Permite crearea de metode în repository care generează automat interogări pe baza numelor metodologice.
- Suport pentru Query-uri Personalizate: Permite scrierea de query-uri personalizate folosind JPQL (Java Persistence Query Language) sau SQL nativ.
- Caching: Hibernate, ca implementare JPA, oferă mecanisme de caching pentru îmbunătățirea performanței aplicațiilor prin reducerea numărului de interogări la baza de date.

Dezavantaje:

- Abstracție Ridicată: Nivelului de abstractizare poate face dificilă rezolvarea erorilor în comparație cu alte metode.
- Performanță: În unele cazuri, performanța poate fi afectată datorită mecanismelor de gestionarea entităților și a tranzacțiilor.

Astfel, am ales să folosesc JPA în special datorită faptului că face o mapare a datelor între obiecte și baza de date relațională automat cu implementarea Hibernate pentru a adăuga și nivelul de caching pentru performanță. De asemenea, suportul pentru Lazy Loading a fost de ajutor în contextul în care unele tabele conțin BLOB-uri, aceste date fiind aduse doar când este necesar.

#### 5.2.4 Implementarea JPA

Pentru a configura o aplicație Spring Boot care folosește JPA și Hibernate cu o bază de date MySQL, este nevoie de câteva setări în fișierul `application.properties`.

```
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/licenta
spring.datasource.username=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.show-sql= true
```

Figură 25: Configurarea conexiunii cu baza de date

Explicația setărilor:

- `spring.jpa.hibernate.ddl-auto=update` Aceasta setare specifică strategia Hibernate pentru actualizarea schemei bazei de date. Opțiunea „update” va actualiza schema

bazei de date la pornirea aplicației, adăugând tabele noi sau modificând tabelele existente pentru a se potrivi cu modelul obiectului Java.

- Restul setărilor țin de conexiunea cu baza de date.

```
9 usages
@Entity
@NoArgsConstructor
@AllArgsConstructor
@Getter
@Builder
@Setter
public class Enroll {

    no usages
    @Id
    @GeneratedValue(strategy = GenerationType.UUID)
    private String enrollmentId;

    no usages
    @ManyToOne
    private User user;

    no usages
    @ManyToOne
    private Course course;

}
```

Figură 26: Exemplu de entitate

Un repository este o componenta ce oferă un mecanism pentru accesarea, manipularea și interogarea bazelor de date. În continuare voi prezenta un exemplu de repository din aplicație.

```
@Repository
public interface EnrollRepository extends JpaRepository<Enroll, String> {

    1 usage
    @Query("SELECT e.course FROM Enroll e WHERE e.user.email = :email")
    List<Course> findCoursesByUserEmail(@Param("email") String email);

    1 usage
    @Transactional
    @Modifying
    @Query("DELETE FROM Enroll e WHERE e.user.userId = :userId AND e.course.courseId = :courseId")
    int deleteByUserIdAndCourseId(@Param("userId") String userId, @Param("courseId") String courseId);

    1 usage
    void deleteByCourse_CourseId(String courseId);

}
```

Figură 27: Un exemplu de Repository

În repository-ul EnrollRepository observăm că, datorită utilizării Spring Data JPA, metoda `deleteByCourse_CourseId(String courseId)` este generată automat de bibliotecă pe baza convențiilor de denumire. În schimb, pentru celelalte metode, definesc explicit interogările folosind adnotarea `@Query`.

### 5.3 Paginare si cautare dupa diferite informatii

O alta problema pe care am intampinat-o in aplicatie a fost paginarea si cautarea. Am folosit paginare pentru lista de exercitii dar si pentru lista de incarcari a temelor de catre studenti.

Problema este ca voiam ca aplicatia sa faca cautarea direct prin cerere in baza de date, nu sa fac cerere apoi sa sortez elementele, pentru ca acest lucru ar fi redus din performanta, de asemenea voiam cautarea pe mai multe campuri in acelasi timp, ceea ce necesita o cerere complexa.

Pentru a rezolva aceasta problema, am implementat urmatoarea solutie:

```
@GetMapping ("/get/assignments")
ResponseBody<?> getAssignments(
    @RequestParam(name = "page", required = false, defaultValue = "0") Long page,
    @ModelAttribute SearchAssignmentsDTO searchAssignmentsDTO) {
    System.out.println(searchAssignmentsDTO);
    long totalRecords = assignmentService.countAssignments(searchAssignmentsDTO);
    long totalPages = (totalRecords + 20 - 1) / 20;
    SearchResultsPageDTO searchResultsPageDTO = new SearchResultsPageDTO();
    List<AssignmentDTO> searchResults = assignmentService.getAllAssignments(searchAssignmentsDTO, page, pageSize: 20L);
    searchResultsPageDTO.setTotalPages(totalPages);
    searchResultsPageDTO.setAssignments(searchResults);
    return ResponseEntity.ok(searchResultsPageDTO);
}
```

Figură 28: Metoda din controller pentru cautarea exercitiilor

1. Expunerea unui REST endpoint care primește de la client numărul paginii si un obiect de tipul `SearchAssignmentsDTO` care specifică criteriile de căutare.

```
@Getter
@Setter
@ToString
public class SearchAssignmentsDTO {
    no usages
    private String assignmentId;
    no usages
    private String category;
    no usages
    private String title;
    no usages
    private String createdBy;
    no usages
    @DateTimeFormat(pattern = "yyyy-MM-dd")
    private Date createDate;
}
```

Figură 29: Clasa cu filtrele cautarii

2. După cum observăm ne folosim de `assignmentService` pentru a primi numărul de exerciții care există în baza de date cu aceste criterii.

```
1 usage
public long countAssignments(SearchAssignmentsDTO dto) {
    return assignmentRepository.countAssignmentsByCriteria(dto);
}
```

Figură 30: Metoda pentru numărarea exercițiilor

```
2 usages
@Repository
public interface AssignmentRepository extends JpaRepository<Assignment, String>, CustomAssignmentRepository {
    1 usage
    @Query("SELECT new com.licenta.StuddyBuddy.dto.AssignmentDTO(c.assignmentId, c.category, c.title, c.createdBy, c.createdDate) FROM Assignment c where c.assignmentId = :assignmentId")
    AssignmentDTO getAssignmentMetadata(String assignmentId);

    1 usage
    @Query("SELECT c.content FROM Assignment c where c.assignmentId = :assignmentId")
    byte[] getAssignmentContent(String assignmentId);

    1 usage
    @Query("SELECT c.solution FROM Assignment c where c.assignmentId = :assignmentId")
    byte[] getAssignmentSolution(String assignmentId);
}
```

Figură 31: Repository-ul pentru exerciții

3. Metoda din repository este adusă de extinderea `CustomAssignmentRepository`.

```
public interface CustomAssignmentRepository {
    1 usage 1 implementation
    List<Assignment> findAssignmentsByCriteria(SearchAssignmentsDTO dto, Long page, Long pageSize);
    1 usage 1 implementation
    long countAssignmentsByCriteria(SearchAssignmentsDTO dto);
}
```

Figură 32: Interfața

### 5.3.1 Detalii de implementare

Astfel putem observa implementarea metodei în clasa `CustomAssignmentRepositoryImpl` care implementează această interfață și suprascrie cele două metode.

```

@Override
public long countAssignmentsByCriteria(SearchAssignmentsDTO dto) {
    CriteriaBuilder cb = entityManager.getCriteriaBuilder();
    CriteriaQuery<Long> query = cb.createQuery(Long.class);
    Root<Assignment> assignment = query.from(Assignment.class);

    List<Predicate> predicates = new ArrayList<>();

    if (dto.getAssignmentId() != null && !dto.getAssignmentId().isEmpty()) {
        predicates.add(cb.equal(assignment.get("assignmentId"), dto.getAssignmentId()));
    }
    if (dto.getCategory() != null && !dto.getCategory().isEmpty()) {
        predicates.add(cb.equal(assignment.get("category"), dto.getCategory()));
    }
    if (dto.getTitle() != null && !dto.getTitle().isEmpty()) {
        predicates.add(cb.like(cb.lower(assignment.get("title")), pattern: "%" + dto.getTitle().toLowerCase() + "%"));
    }
    if (dto.getCreatedBy() != null && !dto.getCreatedBy().isEmpty()) {
        predicates.add(cb.equal(assignment.get("createdBy"), dto.getCreatedBy()));
    }
    if (dto.getCreatedDate() != null) {
        LocalDate createdDateLocal = dto.getCreatedDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate();
        predicates.add(cb.equal(cb.function( name: "date", LocalDate.class, assignment.get("createdDate")), createdDateLocal));
    }

    query.select(cb.count(assignment)).where(predicates.toArray(new Predicate[0]));
    return entityManager.createQuery(query).getSingleResult();
}

```

Figură 33: Implementarea funcției pentru cautarea exercitiilor

În această funcție `countAssignmentsByCriteria(SearchAssignmentsDTO)` se creează un `CriteriaBuilder` pentru a construi interogări dinamice, `CriteriaQuery` este configurat pentru a întoarce `Long` care reprezintă numărul de înregistrări ce se potrivesc criteriilor, `Root` reprezintă tabelul de bază în acest caz `Assignment`, `List<Predicate>` este lista ce conține condițiile interogării, apoi se face un query select care contorizează numărul de exerciții cu predicatele create și returnează rezultatul.

```

int pageSizeValue = (pageSize != null) ? pageSize.intValue() : 20;
int pageNumber = (page != null) ? page.intValue() : 0;
int offset = pageNumber * pageSizeValue;

return entityManager.createQuery(query)
    .setFirstResult(offset)
    .setMaxResults(pageSizeValue)
    .getResultList();

```

Figură 34: implementarea paginării

Funcția `findAssignmentsByCriteria(SearchAssignmentsDTO, Long page, Long pagesize)` este asemănătoare din punct de vedere al predicatelor, având aceleași criterii de căutare setate doar că la acest pas este adăugată paginarea prezentă în figura x. Astfel pentru această cerere trebuie calculat indexul de start pentru interogarea bazei de date și numărul de înregistrări care sunt returnate pe acea pagină, iar acestea sunt date ca parametru în interogare de asemenea.



În final în controller după ce acestea sunt setate este construit un obiect de tipul `SearchResultsPageDTO` care conține numărul total de pagini și lista de exerciții, pentru a fi afișate pe partea de client.

## **6 STUDIUL DE CAZ / EVALUAREA REZULTATELOR**

Acest capitol are ca scop principal evaluarea și analiza soluției dezvoltate. Ne vom concentra pe două aspecte esențiale. În primul rând, vom examina dacă soluția funcționează corect, conform specificațiilor definite în capitolul 2. În al doilea rând, vom evalua performanța acesteia, comparativ cu soluțiile existente, utilizând metrici clare și cuantificabile.

Pentru a măsura performanța și corectitudinea aplicației, ne vom folosi de extensia `Web Vitals` pentru Chrome și de instrumentul `Lighthouse`. Instrumentele acestea ne vor permite să obținem date relevante despre diverse aspecte ale performanței aplicației, inclusiv timpi de încărcare, interactivitate și stabilitate vizuală.

### **6.1 Analiza Web Vitals**

`Web Vitals` este o inițiativă de la Google ce oferă îndrumare dezvoltatorilor pentru ca aceștia să poată să realizeze dacă oferă sau nu o experiență bună utilizatorilor.

`Web Vitals` se focusează pe trei criterii importante:

1. **LCP (Largest Contentful Paint):** Acest criteriu măsoară performanța de încărcare. Pentru o bună experiență aceasta trebuie să se încadreze între următorii parametri: sub 2.5 secunde experiență bună, între 2.5 și 4s este loc de îmbunătățire, iar ce este peste 4s experiență slabă pentru utilizator.
2. **INP (Interaction to Next Paint):** Criteriu ce măsoară interactivitatea. Pentru ca acest criteriu să se încadreze la o experiență bună, trebuie să fie sub 200ms
3. **CLS (Cumulative Layout Shift):** Măsoară stabilitatea vizuală (de exemplu animațiile pe care le are aplicația cât de bine funcționează). Pentru a oferi o experiență bună pentru utilizator aceasta trebuie să fie mai mică decât 0.1.

Astfel analizând aplicația am obținut următoarele rezultate:

1. Pe pagina de autentificare, rezultatul analizei, ne indică ca utilizatorul are parte de o experiență bună, unde locurile de îmbunătățire sunt foarte mici. Putem observa figura x

▶ [Web Vitals Extension] LCP 0.528 secs (good)	vitals.js:262
▶ [Web Vitals Extension] FCP 0.528 secs (good)	vitals.js:262
▶ [Web Vitals Extension] TTFB 0.340 secs (good)	vitals.js:262
▶ [Web Vitals Extension] CLS 0.00 (good)	vitals.js:262
▶ [Web Vitals Extension] FID 2 ms (good)	vitals.js:262
▶ [Web Vitals Extension] INP 8 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 48 ms (good)	vitals.js:262
▶ [Web Vitals Extension] INP 48 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 16 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 24 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 24 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 16 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 24 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	vitals.js:262

Figură 35: Analiza pagina autentificare

2. Pe pagina de administrator o imbunatatire care este posibila ar fi din perspectiva CLS, mai exact incarcarea problemei la editarea acesteia. Putem observa ca indicele CLS este foarte putin peste metrica noastra de experienta buna a utilizatorului, astfel incat nu este un impact foarte mare.

▶ [Web Vitals Extension] LCP 0.486 secs (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] LCP 1.187 secs (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] FCP 0.486 secs (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] TTFB 0.341 secs (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] CLS 0.14 (needs-improvement)	VM1073 vitals.js:262
▶ [Web Vitals Extension] Interaction 16 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] FID 3 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] INP 16 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] Interaction 24 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] Interaction 40 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] INP 40 ms (good)	VM1073 vitals.js:262
▶ [Web Vitals Extension] Interaction 24 ms (good)	VM1073 vitals.js:262

Figură 36: Analiza pagina administrator

3. Pe pagina ce arata cursurile la care este înrolat atât studentul, cat si profesorul metricile sunt exprima ca utilizatorul are parte de o experienta buna.

▶ [Web Vitals Extension] LCP 0.504 secs (good)	vitals.js:262
▶ [Web Vitals Extension] LCP 0.771 secs (good)	vitals.js:262
▶ [Web Vitals Extension] LCP 0.838 secs (good)	vitals.js:262
▶ [Web Vitals Extension] FCP 0.504 secs (good)	vitals.js:262
▶ [Web Vitals Extension] TTFB 0.338 secs (good)	vitals.js:262
▶ [Web Vitals Extension] CLS 0.00 (good)	vitals.js:262
▶ [Web Vitals Extension] LCP 0.905 secs (good)	vitals.js:262

Figură 37: Analiza pagina cu toate cursurile la care un utilizator este inrolat

4. Pagina cursurilor de asemenea ofera utilizatorului o experienta buna.

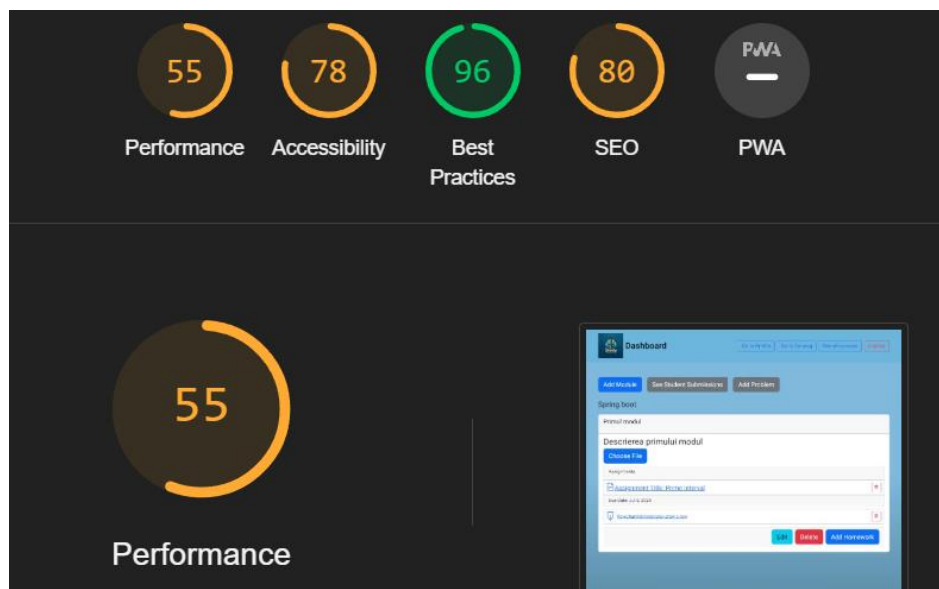
▶ [Web Vitals Extension] LCP 0.490 secs (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] LCP 0.807 secs (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] LCP 0.840 secs (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] FCP 0.490 secs (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] TTFB 0.349 secs (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] CLS 0.00 (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] Interaction 40 ms (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] FID 2 ms (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] INP 40 ms (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] Interaction 40 ms (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] Interaction 32 ms (good)	VM2399_vitals.js:262
▶ [Web Vitals Extension] CLS 0.03 (good)	VM2399_vitals.js:262

Figură 38: Analiza pagina cursului

Astfel analiza Web Vitals arata ca majoritatea paginilor din aplicatie ofera o experienta buna utilizatorilor. Putem observa ca paginile se incadreaza in parametrii recomandati de Google pentru o experienta optima. Cu toate acestea editarea problemelor poate fi ajustata minor pentru a imbunatatii stabilitatea vizuala.

## 6.2 Analiza Lighthouse

Lighthouse este un instrument dezvoltat de Google, asemanator Web Vitals, acesta analizeaza performanta, accesibilitatea si daca s-au respectat cele mai bune practici in dezvoltare.



Figură 39: Analiza Light House

Am efectuat o analiza detaliata a paginilor din aplicatia noastra, dar ne vom concentra in special pe pagina care v-a fi cea mai folosita, pagina unui curs.

Rezultatele analizei si locurile de imbunatatire:

1. Activarea compresiei textului: Activarea compresiei textului poate reduce dimensiunea paginilor web și, astfel, poate îmbunătăți timpii de încărcare.
2. Largest Contentful Paint (LCP) element: Acest aspect se referă la sigla aplicației. Îmbunătățirea timpii de încărcare a acestui element poate reduce LCP și poate îmbunătăți experiența utilizatorilor.
3. Evitarea codului legacy JavaScript: Aceasta se referă la o clasă importată de Angular, ce nu este folosită, astfel aceasta nu impactează deloc experiența utilizatorilor.

Rezultatele analizelor efectuate atât cu Web Vitals, cât și cu Lighthouse, indică faptul că aplicația noastră oferă o experiență bună utilizatorilor. Cu toate acestea, există câteva locuri de îmbunătățire. Problemele identificate vor impacta semnificativ experiența utilizatorilor, dar optimizările sugerate vor contribui la o experiență și mai bună.

În plus, o viziune pe care o am este migrarea fișierelor la Firebase. Rezultatul acestei schimbări va soluționa problema identificată prin analiza cu aceste instrumente, referitoare la LCP, deoarece Firebase permite o livrare mai rapidă și mai eficientă a resurselor. În concluzie, implementarea acestor optimizări va asigura că aplicația noastră nu doar îndeplinește, ci depășește așteptările utilizatorilor în ceea ce privește performanța și calitatea.

## 7 CONCLUZII

În acest capitol este sumarizat întreg proiectul, de la obiective, la implementare, și la relevanța rezultatelor obținute. În finalul capitolului poate exista o subsecțiune de „Dezvoltări ulterioare”.

Criterii pentru calificativul *Nesatisfăcător*:

- Concluziile nu sunt corelate cu conținutul lucrării;

Criterii pentru calificativul *Satisfăcător*:

- Concluziile sunt corelate cu conținutul lucrării, însă nu se oferă o imagine asupra calității și relevanței rezultatelor obținute;

Criterii pentru calificativul *Bine*:

- Concluziile sunt corelate cu conținutul lucrării, și se oferă o imagine precisă asupra relevanței și calității rezultatelor obținute în cadrul proiectului.

## 8 BIBLIOGRAFIE

- Trebuie respectat **un singur standard** de trimiteri bibliografice (citare), **dintre** următoarele alternative:
  - APA (<http://pitt.libguides.com/c.php?g=12108&p=64730>)
  - IEEE (<https://iee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf>)
  - Harvard (<https://libweb.anglia.ac.uk/referencing/harvard.htm>)
  - Cu numerotarea referințelor în ordine alfabetică sau în ordinea apariției în text (de exemplu, stilul cu numere folosit de unele publicații ACM - <https://www.acm.org/publications/authors/reference-formatting>)
- Toate referințele din acest capitol trebuie să fie referite în text. Exemple:
  - [Articol jurnal]: [2];
  - [Articol conferință]: [3];
  - [Carte]: [4]
  - [Weblink]: [5]
  - [Application report] [6]

- [1] J. Silva-Martinez, "ELEN-325. Introduction to Electronic Circuits: A Design Approach," 2008. [Online]. Available: <http://www.ece.tamu.edu/~spalermo/ecen325/Section%20III.pdf>.
- [2] H. Baali, H. Djelouat, A. Amira and F. Bensaali, "Empowering Technology Enabled Care Using IoT and Smart Devices: A Review," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 1790-1809, 2018.
- [3] A. Haroon, S. Akram, M. A. Shah and A. Wahid, "E-Lithe: A Lightweight Secure DTLS for IoT," in *IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Toronto, 2017.
- [4] A. K. Jain and R. C. Dubes., *Algorithms for Clustering Data*, Upper Saddle River: Prentice-Hall, Inc., 1988.
- [5] "Kernel panic! What are Meltdown and Spectre, the bugs affecting nearly every computer and device?," *techcrunch.com*, 2018. [Online]. Available: <https://techcrunch.com/2018/01/03/kernel-panic-what-are-meltdown-and-spectre-the-bugs-affecting-nearly-every-computer-and-device>. [Accessed 14 02 2018].
- [6] E. Rogers, "Understanding Buck-Boost Power Stages in Switch Mode Power Supplies," Texas Instruments, 2007.

- NU utilizați referințe la Wikipedia sau alte surse fără autor asumat.
- Pentru referințe la articole relevante accesibile în web (descrise prin URL) se va nota la bibliografie și data accesării.
- Mai multe detalii despre citarea referințelor din internet se pot regăsi la:
  - <http://www.writinghelp-central.com/apa-citation-internet.html>
  - <http://www.webliminal.com/search/search-web13.html>
- Note de subsol se utilizează dacă referiți un link mai puțin semnificativ o singură dată; Dacă nota este citată de mai multe ori, atunci utilizați o referință bibliografică.
- Dacă o imagine este introdusă în text și nu este realizată de către autorul lucrării, trebuie citată sursa ei (ca notă de subsol sau referință - este de preferat utilizarea unei note de subsol).
- Referințele se pun direct legate de text (de exemplu „KVM [1] uses”, „as stated by Popescu and Ionescu [12]”, etc.). Nu este recomandat să folosiți formulări de tipul „[1] uses”, „as stated in [12]”, „as described in [11]” etc.
- Afirmatiile de forma „are numerous”, „have grown exponentially”, „are among the most used”, „are an important topic” trebuie să fie acoperite cu citări, date concrete și analize comparative.
  - Mai ales în capitolele de introducere, „state of the art”, „related work” sau „background” trebuie să vă argumentați afirmațiile prin citări. Fiți autocritici și gândiți-vă dacă afirmațiile au nevoie de citări, chiar și cele pe care le considerați evidente.
  - Cea mai mare parte dintre citări vor fi în capitolele de introducere „state of the art”, „related work” sau „background”.
- Toate intrările bibliografice trebuie citate în text. Nu le adăugați pur și simplu la final.
- Nu copiați sau traduceți niciodată din surse de informație de orice tip (online, offline, cărți, etc.). Dacă totuși doriți să oferiți, prin excepție, un citat celebru - de maxim 1 frază- utilizați ghilimele și evident menționați sursa.
- Dacă reformulați idei sau creați un paragraf rezumat al unor idei folosind cuvintele voastre, precizați cu citare (referință bibliografică) sau cu notă de subsol sursa sau sursele de unde ați preluat ideile.

## 9 ANEXE

Anexele sunt opționale.

Ce poate intra în anexe:

- Exemplu de fișier de configurare sau compilare;
- Un tabel mai mare de ½ pagină;
- O figura mai mare de ½ pagină;
- O secvență de cod sursă mai mare de ½ pagină;
- Un set de capturi de ecran („screenshot”-uri);
- Un exemplu de rulare a unor comenzi plus rezultatul („output”-ul) acestora;
- În anexe intră lucruri care ocupă mai mult de o pagină ce ar întrerupe firul natural de parcurgere al textului.